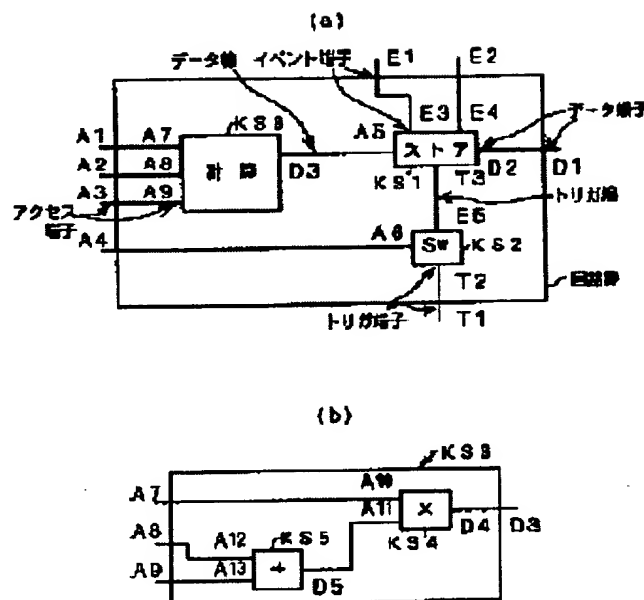


DEVICE AND METHOD FOR GENERATING PROGRAM

Patent number: JP10116184
Publication date: 1998-05-06
Inventor: KAJIOKA SHIGERU; MURAYAMA HIROYUKI; ITO KENZO; FUJII TOMOYASU
Applicant: DENSO CORP.; DENSO CREATE KK
Classification:
- international: G06F9/06; G06F11/28
- european:
Application number: JP19960270202 19961011
Priority number(s): JP19960270202 19961011

Abstract of JP10116184

PROBLEM TO BE SOLVED: To generate a desired program efficiently and automatically.
SOLUTION: A virtual circuit element KS which visually represents the contents of a process is equipped with a data base. As for more than one kind of element, i.e., a circuit element having at least E or D terminals among event E (E1 to E5) terminals which receive a trigger signal and return a completion signal for corresponding operation, trigger (T1 to T3) terminals which have the opposite functions from the D terminals, data D (D1 to D5) terminals which receive an access signal and return corresponding data, and access A (A1 to A13) terminals having the opposite functions from the D terminals, this data base is stored with a small program and terminal information representing the elements. A circuit diagram drawn by connecting circuit elements KS (KS1 to KS5) is read in and respective circuit elements are extracted from the circuit diagram; and the small program corresponding to each circuit element is specified by referring to the data base, and respective small programs are merged according to the connection state based upon the terminal functions of the circuit elements to generate a program of process contents that the circuit diagram shows.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平10-116184

(43)公開日 平成10年(1998)5月6日

(51)Int.Cl.⁶

G 0 6 F 9/06

識別記号

5 3 0

11/28

3 4 0

F I

G 0 6 F 9/06

11/28

5 3 0 G

5 3 0 Q

3 4 0 C

審査請求 未請求 請求項の数13 O L (全 44 頁)

(21)出願番号

特願平8-270202

(22)出願日

平成8年(1996)10月11日

(71)出願人 00004260

株式会社デンソー

愛知県刈谷市昭和町1丁目1番地

(71)出願人 396021944

株式会社デンソークリエイト

愛知県名古屋市中区錦二丁目15番20号

(72)発明者 梶岡 繁

愛知県刈谷市昭和町1丁目1番地 株式会
社デンソー内

(72)発明者 村山 浩之

愛知県刈谷市昭和町1丁目1番地 株式会
社デンソー内

(74)代理人 弁理士 足立 勉

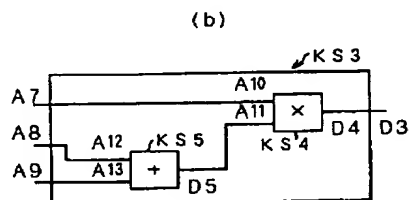
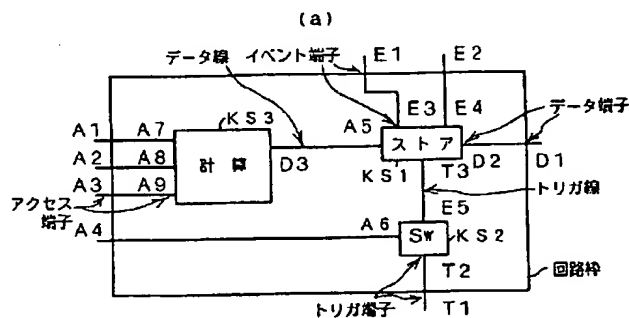
最終頁に続く

(54)【発明の名称】 プログラムを作成するための装置及び媒体

(57)【要約】

【課題】 所望のプログラムを効率良く自動作成できる装置を提供する。

【解決手段】 処理の内容を視覚的に表す仮想の回路素子KSであって、トリガ信号を受けて対応する動作の完了信号を返すイベント(E)端子と、該E端子と反対機能のトリガ(T)端子と、アクセス信号を受けて対応するデータを返すデータ(D)端子と、該D端子と反対機能のアクセス(A)端子とのうち、少なくともE又はD端子を有する回路素子の情報として、当該素子の処理内容を表す小プログラムと端子情報とを、複数種類の素子について記憶したデータベースを備え、複数の回路素子KSを結線して描画された回路図を読み込んで該回路図から各回路素子を抽出し、上記データベースを参照して各回路素子に対応する小プログラムを特定すると共に、該各小プログラムを回路素子同士の端子機能に基づく結線状態に従い結合して、上記回路図が表す処理内容のプログラムを生成する。



【特許請求の範囲】

【請求項 1】 コンピュータに実行させる処理の内容を所定の表示媒体上で視覚的に表現する仮想回路素子であって、前記処理の実行開始を指令する駆動信号を外部から受けるための第 1 の端子と、前記処理の実行により他の仮想回路素子へ前記駆動信号を発するのための第 2 の端子と、前記処理の実行で得られるデータに対する要求信号を外部から受けるための第 3 の端子と、前記処理の実行により他の仮想回路素子へ前記要求信号を発するための第 4 の端子との 4 種類の端子のうち、少なくとも前記第 1 又は第 3 の端子を有する仮想回路素子の情報として、該仮想回路素子の処理内容を表す小プログラムと、該仮想回路素子が有する端子及びその種類を表わす端子情報とを、複数種類の仮想回路素子について記憶した素子情報記憶手段を備え、該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図を解析することで、該疑似回路図により表現されたプログラムを作成するプログラム作成装置であって、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、当該疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の結合関係を求め、該結合関係に従って前記各小プログラムを結合したプログラムを構築するプログラム構築手段と、を備えたことを特徴とするプログラム作成装置。

【請求項 2】 コンピュータに実行させる処理の内容を所定の表示媒体上で視覚的に表現する仮想回路素子であって、前記処理の実行開始を指令する駆動信号を外部から受けると共に該処理の実行完了を表わす完了信号を前記駆動信号の発信先へ返すための第 1 の端子と、前記処理の実行により他の仮想回路素子へ前記駆動信号を発して該他の仮想回路素子から前記完了信号を受けるための第 2 の端子と、前記処理の実行で得られるデータに対する要求信号を外部から受けると共に該データを前記要求信号の発信先へ返すための第 3 の端子と、前記処理の実行により他の仮想回路素子へ前記要求信号を発して該他の仮想回路素子からデータを受けるための第 4 の端子との 4 種類の端子のうち、少なくとも前記第 1 又は第 3 の端子を有する仮想回路素子の情報として、該仮想回路素子の処理内容を表す小プログラムと、該仮想回路素子が有する端子及びその種類を表わす端子情報とを、複数種類の仮想回路素子について記憶した素子情報記憶手段を備え、

該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図を解析することで、該疑似回路図により表現されたプログラムを作成するプログラム作成装置であって、

前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、当該疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の結合関係を求め、該結合関係に従って前記各小プログラムを結合したプログラムを構築するプログラム構築手段と、

を備えたことを特徴とするプログラム作成装置。

【請求項 3】 請求項 1 又は請求項 2 に記載のプログラム作成装置において、

作業による操作に従って前記疑似回路図を描画するための描画手段を備え、前記読込手段は、前記描画手段により描画された疑似回路図を読み込むこと、

を特徴とするプログラム作成装置。

【請求項 4】 請求項 1 ないし請求項 3 の何れかに記載のプログラム作成装置において、

前記疑似回路図は、所定の表示装置に当該疑似回路図を表示するための情報と、当該疑似回路図中の仮想回路素子、端子、及び該端子の接続を特定するための索引情報とにより構成され、

前記素子情報記憶手段は、前記小プログラム及び前記端子情報を、前記索引情報によって特定できるように記憶しており、

前記読込手段は、前記索引情報を読み込んで、該索引情報を前記プログラム構築手段に与え、

前記プログラム構築手段は、前記索引情報に従い前記素子情報記憶手段を参照することによって、前記疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を解析して、前記特定される各小プログラム同士の結合関係を求め、該結合関係に従って前記各小プログラムを結合したプログラムを構築すること、

を特徴とするプログラム作成装置。

【請求項 5】 請求項 1 ないし請求項 4 の何れかに記載のプログラム作成装置において、

前記素子情報記憶手段には、前記小プログラムとして、所定の処理を表す関数が記憶されており、前記プログラム構築手段は、前記疑似回路図により表現されたプログラムを C 言語で構築すること、

を特徴とするプログラム作成装置。

【請求項 6】 請求項 5 に記載のプログラム作成装置において、前記小プログラムとして前記素子情報記憶手段に記憶された関数は、C 言語で記載されたプログラムであること、を特徴とするプログラム作成装置。

【請求項 7】 請求項 1 又は請求項 2 に記載の素子情報記憶手段を備え、該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線した疑似回路図を描画するための描画装置であって、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を、所定の表示装置の表示画面における所定領域に一覧表示させる一覧表示手段と、前記所定領域に表示された仮想回路素子の中から、作業による操作に従って何れかの仮想回路素子を選択し、該選択した仮想回路素子を、前記表示装置の表示画面において前記所定領域とは異なる領域として設定された編集領域に表示させる素子選択手段と、該素子選択手段により前記編集領域に表示された複数の仮想回路素子間を、作業による操作に従って信号線により該編集領域上で結線する結線手段と、前記編集領域に表示された仮想回路素子及び信号線からなる疑似回路図を所定の記憶媒体に記録させる記録制御手段と、を備えたことを特徴とする描画装置。

【請求項 8】 請求項 1 又は請求項 2 に記載の素子情報記憶手段を備え、該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図によって表現される処理内容を検証するためのシミュレーション装置であって、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた前記疑似回路図を、所定の表示装置の表示画面上に表示させる疑似回路図表示手段と、前記表示された疑似回路図中の何れかの仮想回路素子が有する前記第 1 又は第 3 の端子を、作業による操作に従って選択する端子選択手段と、前記表示された疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、前記各仮想回路素子に対応する小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の実行順序関係を求め、前記特定される各小プログラムの表す処理内容を、前記求めた実行順序関係に従って、前記端子選択手段により前記第 1 又は第 3 の端子が選択された仮想回路素子に対応する処理内容から順次実

行するプログラム実行手段と、

前記表示された疑似回路図中の仮想回路素子のうち、前記プログラム実行手段による実行結果をモニタしたい処理内容に対応する仮想回路素子が有する端子を、作業による操作に従って選択するモニタ対象選択手段と、該モニタ対象選択手段により端子が選択された仮想回路素子が表す処理内容の前記プログラム実行手段による実行結果を、前記表示装置の表示画面上に表示させる実行結果表示手段と、

を備えたことを特徴とするシミュレーション装置。

【請求項 9】 請求項 1 又は請求項 2 に記載の素子情報記憶手段を備えたコンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図を解析することで、該疑似回路図により表現されたプログラムを作成するプログラム作成装置として機能させるための処理プログラムを記録した機械読み取り可能な媒体であって、

前記記録される処理プログラムは、コンピュータを、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、当該疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の結合関係を求め、該結合関係に従って前記各小プログラムを結合したプログラムを構築するプログラム構築手段と、して機能させるためのプログラムであることを特徴とする媒体。

【請求項 10】 請求項 1 又は請求項 2 に記載の素子情報記憶手段を備えたコンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線した疑似回路図を描画するための描画装置として機能させるための処理プログラムを記録した機械読み取り可能な媒体であって、

前記記録される処理プログラムは、コンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を、所定の表示装置の表示画面における所定領域に一覧表示させる一覧表示手段と、前記所定領域に表示された仮想回路素子の中から、作業による操作に従って何れかの仮想回路素子を選択し、該選択した仮想回路素子を、前記表示装置の表示画面において前記所定領域とは異なる領域として設定された編集領域に表示させる素子選択手段と、該素子選択手段により前記編集領域に表示された複数の仮想回路素子間を、作業による操作に従って信号線に

より該編集領域上で結線する結線手段と、
前記編集領域に表示された仮想回路素子及び信号線からなる疑似回路図を所定の記憶媒体に記録させる記録制御手段と、して機能させるためのプログラムであることを特徴とする媒体。

【請求項 11】 請求項 1 又は請求項 2 に記載の素子情報記憶手段を備えたコンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図によって表現される処理内容を検証するためのシミュレーション装置として機能させるための処理プログラムを記録した機械読み取り可能な媒体であって、前記記録される処理プログラムは、コンピュータを、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた前記疑似回路図を、所定の表示装置の表示画面上に表示させる疑似回路図表示手段と、
前記表示された疑似回路図中の何れかの仮想回路素子が有する前記第 1 又は第 3 の端子を、作業者による操作に従って選択する端子選択手段と、
前記表示された疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、前記各仮想回路素子に対応する小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の実行順序関係を求め、前記特定される各小プログラムの表す処理内容を、前記求めた実行順序関係に従って、前記端子選択手段により前記第 1 又は第 3 の端子が選択された仮想回路素子に対応する処理内容から順次実行するプログラム実行手段と、
前記表示された疑似回路図中の仮想回路素子のうち、前記プログラム実行手段による実行結果をモニタしたい処理内容に対応する仮想回路素子が有する端子を、作業者による操作に従って選択するモニタ対象選択手段と、
該モニタ対象選択手段により端子が選択された仮想回路素子が表す処理内容の前記プログラム実行手段による実行結果を、前記表示装置の表示画面上に表示させる実行結果表示手段と、して機能させるためのプログラムであることを特徴とする媒体。

【請求項 12】 請求項 1 ないし請求項 6 の何れかに記載のプログラム作成装置において、
前記プログラム構築手段は、
前記疑似回路図により表現されたプログラムの各機能部分として用いられる前記各小プログラムを結合する際に、データのやり取りを行う 2 つの小プログラム間にて夫々扱われるデータの形態が異なっているか否かを判定し、前記両データの形態が異なっている場合には、前記両データの形態を一致させるためのデータ形態変換用プログラムを作成して前記構築するプログラムに加えるこ

と、
を特徴とするプログラム作成装置。

【請求項 13】 請求項 1 ないし請求項 6 及び請求項 12 の何れかに記載のプログラム作成装置において、
当該装置により作成されるプログラムは、内燃機関を制御するための制御プログラムであること、
を特徴とするプログラム作成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プログラムを作成するための装置に関する。

【0002】

【従来の技術】従来より、所望の制御システムを実現するためのプログラムを作成する場合には、まず、作成しようとするプログラムの処理内容（換言すれば、ソフトウェアの仕様）をフローチャートで表現し、そのフローチャートに沿って、所定のプログラミング言語によるプログラムを作成する（所謂、コーディングを行う）ようにしている。

【0003】

【発明が解決しようとする課題】しかしながら、ソフトウェアの仕様をフローチャートによって表現する従来の方式では、作成しようとするプログラムにおける各処理同士の関係や、その各処理間でのデータ授受の関係が総合的に理解し難く、このため、プログラムを効率良く作成することができなかった。即ち、フローチャートによってソフトウェアの仕様を表現する場合には、ある処理の実行中に他の処理へサブルーチンジャンプしたり、ある処理の実行によって演算したデータを他の処理で参照する、といった処理の流れが理解し難いのである。

【0004】そして、このようなことから、フローチャートを描いてプログラムを作成するには、ソフトウェア及びプログラミング言語に関して十分な知識が必要であった。一方、所定の規則に従ってフローチャートを描くと、そのフローチャートの表す処理内容に応じたプログラムを所定のプログラミング言語で自動的に生成する、といったプログラム作成装置も考えられるが、このような装置でも、作成しようとするプログラムをフローチャートによって表さなければならず、上記問題を解決することはできない。

【0005】本発明は、このような問題に鑑みなされたものであり、ソフトウェアに関する知識に精通していなくても所望のプログラムを効率良く作成することのできる装置を提供することを目的としている。

【0006】

【課題を解決するための手段、及び発明の効果】上記目的を達成するためになされた請求項 1 に記載の本発明は、コンピュータに実行させる処理の内容を所定の表示媒体上で視覚的に表現する仮想回路素子であって、前記処理の実行開始を指令する駆動信号を外部から受けるた

めの第1の端子と、前記処理の実行により他の仮想回路素子へ前記駆動信号を発するのための第2の端子と、前記処理の実行で得られるデータに対する要求信号を外部から受けるための第3の端子と、前記処理の実行により他の仮想回路素子へ前記要求信号を発するのための第4の端子との4種類の端子のうち、少なくとも前記第1又は第3の端子を有する仮想回路素子の情報として、該仮想回路素子の処理内容を表す小プログラムと、該仮想回路素子が有する端子及びその種類を表わす端子情報とを、複数種類の仮想回路素子について記憶した素子情報記憶手段を備え、該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図を解析すること、で、該疑似回路図により表現されたプログラムを作成するプログラム作成装置であって、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、当該疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の結合関係を求め、該結合関係に従って前記各小プログラムを結合したプログラムを構築するプログラム構築手段と、を備えたことを特徴としている。

【0007】このプログラム作成装置によってプログラムを生成するには、まず、素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数種類の仮想回路素子群から、所望の処理内容を表す仮想回路素子を適宜選び、ある仮想回路素子の第2の端子と他の仮想回路素子の第1の端子とを信号線により接続し、また、ある仮想回路素子の第4の端子と他の仮想回路素子の第3の端子とを信号線により接続するという具合に、上記選んだ各仮想回路素子を信号線により結線して疑似回路図を描画しておく。

【0008】次に、上記描画した疑似回路図を読込手段に読み込ませて、プログラム構築手段を動作させると、このプログラム構築手段が、素子情報記憶手段の記憶内容を参照しながら、疑似回路図中の仮想回路素子に対応する小プログラムを特定し、同じく素子情報記憶手段の記憶内容を参照しながら、各仮想回路素子の端子間を接続する信号線を追跡し、信号の授受の順番に基づき各小プログラムの結合関係を決定して、疑似回路図全体が表しているプログラムを所定のプログラミング言語で表現したプログラムを構築する。

【0009】例えば、プログラム構築手段は、疑似回路図を構成する仮想回路素子のうち、第1の端子又は第3の端子が他の仮想回路素子に接続されていない仮想回路素子に対応する小プログラムが最初に起動されるように

し、次に、その小プログラムの実行時に、該小プログラムに対応する仮想回路素子の第2の端子又は第4の端子に接続された仮想回路素子に対応する小プログラムが呼び出されて起動されるようにし、次に、その呼び出されて起動される小プログラムに対応する仮想回路素子の第2の端子又は第4の端子に接続された仮想回路素子に対応する小プログラムが呼び出されて起動されるようにする、・・・、といった解析ルールに従って、疑似回路図中の各仮想回路素子に夫々対応する各小プログラムを結合することにより、上記疑似回路図により表現されたプログラムを構築する。

【0010】そして、これにより、疑似回路図において駆動信号が伝達されると見なされる順、或いは、データに対する要求信号が伝達されると見なされる順に、各仮想回路素子に対応する小プログラムが起動される、といったプログラムが生成される。

【0011】このように本発明のプログラム作成装置によれば、作成しようとするプログラムにおける各処理同士の関係や、各処理間でのデータ授受の関係を、2次元に見ることができ上記各関係を総合的に理解し易い疑似回路図を描画することによって、所望のプログラムを自動的に作成することができるため、ソフトウェアに関する知識に精通していなくても所望のプログラムを効率良く作成することができるようになる。

【0012】一方、請求項2に記載のプログラム作成装置は、請求項1に記載の装置と同様の素子情報記憶手段、読込手段、及びプログラム構築手段を備えているが、素子情報記憶手段に小プログラム及び端子情報が記憶される仮想回路素子が、処理の実行開始を指令する駆動信号を外部から受けると共に該処理の実行完了を表わす完了信号を前記駆動信号の発信先へ返すための第1の端子と、前記処理の実行により他の仮想回路素子へ前記駆動信号を発して該他の仮想回路素子から前記完了信号を受けるための第2の端子と、前記処理の実行で得られるデータに対する要求信号を外部から受けると共に該データを前記要求信号の発信先へ返すための第3の端子と、前記処理の実行により他の仮想回路素子へ前記要求信号を発して該他の仮想回路素子からデータを受けるための第4の端子との4種類の端子のうち、少なくとも前記第1又は第3の端子を有している。

【0013】つまり、請求項1に記載のプログラム作成装置では、仮想回路素子の端子が、駆動信号又はデータに対する要求信号を発するか受けるという一方だけの役割を持っているのであるが、請求項2に記載のプログラム作成装置では、仮想回路素子の端子が、第1の端子であれば駆動信号を受けて完了信号を返し、第2の端子であれば駆動信号を発して完了信号を受け、第3の端子であれば要求信号を受けてデータを返し、第4の端子であれば要求信号を発してデータを受けるというように、信号を双方向にやり取りする役割を持っている。

【0014】そして、このような請求項2に記載のプログラム作成装置によれば、作成しようとするプログラムにおける各処理同士の関係や、各処理間でのデータ授受の関係を、より一層理解し易い疑似回路図を描画することによって、所望のプログラムを自動的に作成することができるようになる。

【0015】ところで、請求項1又は請求項2に記載のプログラム作成装置において、疑似回路図は、別途作成しておいて読込手段に読み込ませるようにしても良いが、請求項3に記載のように、当該プログラム作成装置に、作業者による操作に従って前記疑似回路図を描画するための描画手段を設け、この描画手段により描画された疑似回路図を、前記読込手段が読み込むようにしても良い。そして、このようにすれば、疑似回路図の作成段階からプログラムの構築段階へ速やかに移行することができ、所望のプログラムをより早く作成することができるようになる。

【0016】一方、請求項1ないし請求項3の何れかに記載のプログラム作成装置において、読込手段としては、例えば、紙などの表示媒体上に描画した画像を直接読み取り可能なイメージスキャナの如き機能を有したものをを用いても良いが、この場合には、装置が大がかりなものとなってしまう。

【0017】そこで、請求項4に記載のようにすれば良い。即ち、請求項4に記載のプログラム作成装置では、前記疑似回路図が、所定の表示装置に当該疑似回路図を表示するための情報と、当該疑似回路図中の仮想回路素子、端子、及び該端子の接続を特定するための索引情報とにより構成されており、これに対応して、素子情報記憶手段は、前記小プログラム及び前記端子情報を、前記索引情報によって特定できるように記憶している。

【0018】そして、前記読込手段は、前記索引情報を読み込んで、その索引情報を前記プログラム構築手段に与え、プログラム構築手段は、前記索引情報に従い前記素子情報記憶手段を参照することによって、前記疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を解析して、前記特定される各小プログラム同士の結合関係を求め、その求めた結合関係に従って前記各小プログラムを結合したプログラムを構築する。

【0019】この請求項4に記載のプログラム作成装置によれば、疑似回路図を構成する情報のうち、該疑似回路図を表示するための情報を用いて、プログラムの生成対象となる疑似回路図を所定の表示装置に表示させることができ、また、読込手段及びプログラム構築手段には、画像を処理するための機能を設ける必要が無い。よって、プログラムの生成対象である疑似回路図を表示装置に表示させて、作業者に対し、これから生成されるプログラムの処理内容を確認させることができ、しかも、

画像処理を行うための特別なハードウェアを用いることなく、所望のプログラムを効率良く作成することができるようになる。

【0020】一方また、本発明のプログラム作成装置において、請求項5に記載の如く、前記素子情報記憶手段に、前記小プログラムとして、所定の処理を表す関数を記憶させておき、前記プログラム構築手段が、前記疑似回路図により表現されたプログラムをC言語で構築するように構成すれば、プログラム構築のための処理動作が非常に簡単なものとなる。

【0021】つまり、C言語によるプログラムでは、所定の処理を表す関数から他の処理を表す関数を呼び出すという記述が基本となるため、前記4種類の端子による各仮想回路素子間の結線状態（即ち、前記第1の端子と第2の端子との関係、及び、前記第3の端子と第4の端子との関係）によるプログラムの表現方式と共通点が多く、本発明の適用に適しているからである。

【0022】そして更に、請求項6に記載のように、請求項5に記載のプログラム作成装置において、前記小プログラムとして素子情報記憶手段に記憶される関数を、C言語で記載しておくようにすれば、疑似回路図により表現されたプログラムをC言語で構築するための処理動作を、より一層簡単なものにすることができる。つまり、各小プログラムを結合するための処理が極めて簡単になるからである。

【0023】以上のようなプログラム作成装置に対して、疑似回路図を与えるためには、次のような描画装置が有効である。即ち、具体的には請求項7に記載したように、請求項1又は請求項2に記載の素子情報記憶手段を備え、該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線した疑似回路図を描画するための描画装置であって、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を、所定の表示装置の表示画面における所定領域に一覧表示させる一覧表示手段と、前記所定領域に表示された仮想回路素子の中から、作業者による操作に従って何れかの仮想回路素子を選択し、該選択した仮想回路素子を、前記表示装置の表示画面において前記所定領域とは異なる領域として設定された編集領域に表示させる素子選択手段と、該素子選択手段により前記編集領域に表示された複数の仮想回路素子間を、作業者による操作に従って信号線により該編集領域上で結線する結線手段と、前記編集領域に表示された仮想回路素子及び信号線からなる疑似回路図を所定の記憶媒体に記録させる記録制御手段と、を備えた描画装置を用いると良い。

【0024】この描画装置によって疑似回路図を描画するには、まず、一覧表示手段により表示装置の表示画面における所定領域に一覧表示された複数の仮想回路素子の中から、素子選択手段により、所望の処理内容を表す

仮想回路素子を適宜選択して表示装置の表示画面における編集領域に表示させる。次に、結線手段によって、上記編集領域に表示された複数の仮想回路素子間（詳しくは、第1の端子と第2の端子との間、或いは、第3の端子と第4の端子との間）を信号線により上記編集領域上で結線する。

【0025】そして、このように表示装置の表示画面における編集領域上で描画された疑似回路図は、記録制御手段により所定の記憶媒体に記録されるため、その記録された疑似回路図を、前述したプログラム作成装置の読込手段に読み込ませれば良いのである。

【0026】そして更に、上記編集領域上で描画された疑似回路図に対応して、請求項4に記載の索引情報（即ち、当該疑似回路図中の仮想回路素子、端子、及び該端子の接続を特定するための情報）を生成する手段を追加して設ければ、請求項4に記載のプログラム作成装置に対して疑似回路図を与えることができ、プログラムをより一層効率良く作成可能な装置を提供することができるようになる。

【0027】一方また、このように疑似回路図を描画してからプログラムの生成作業を行う前の段階で、その描画した疑似回路図が表す処理内容を予め検証するためには、次のようなシミュレーション装置が有効である。即ち、具体的には請求項8に記載したように、請求項1又は請求項2に記載の素子情報記憶手段を備え、該素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図によって表現される処理内容を検証するためのシミュレーション装置であって、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた前記疑似回路図を、所定の表示装置の表示画面上に表示させる疑似回路図表示手段と、前記表示された疑似回路図中の何れかの仮想回路素子が有する前記第1又は第3の端子を、作業による操作に従って選択する端子選択手段と、前記表示された疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、前記各仮想回路素子に対応する小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の実行順序関係を求め、前記特定される各小プログラムの表す処理内容を、前記求めた実行順序関係に従って、前記端子選択手段により前記第1又は第3の端子が選択された仮想回路素子に対応する処理内容から順次実行するプログラム実行手段と、前記表示された疑似回路図中の仮想回路素子のうち、前記プログラム実行手段による実行結果をモニタしたい処理内容に対応する仮想回路素子が有する端子を、作業による操作に従って選択するモニタ対象選択手段と、該モニタ対象選択手段により端子が選択された仮想回路素子が表す処理内容

の前記プログラム実行手段による実行結果を、前記表示装置の表示画面上に表示させる実行結果表示手段と、を備えたシミュレーション装置を用いると良い。

【0028】このシミュレーション装置によって疑似回路図が表す処理内容を検証するには、まず、描画した疑似回路図を読込手段に読み込ませる。すると、疑似回路図表示手段により、その疑似回路図が表示装置の表示画面上に表示されるため、次に、その表示された疑似回路図中の何れかの仮想回路素子が有する第1又は第3の端子を、端子選択手段によって選択し、また、上記表示された疑似回路図中の仮想回路素子のうち、その処理内容の実行結果をモニタしたい仮想回路素子の何れかの端子を、モニタ対象選択手段によって選択する。

【0029】そして、プログラム実行手段を作動させると、このプログラム実行手段が、素子情報記憶手段の記憶内容を参照しながら、疑似回路図中の各仮想回路素子に対応する小プログラムを特定すると共に、各仮想回路素子の端子間を接続する信号線を追跡し、各小プログラムの表す処理内容を、信号の授受の順番に基づき、端子選択手段により第1又は第3の端子が選択された仮想回路素子に対応する処理内容から順次実行する。

【0030】すると、モニタ対象選択手段により端子が選択された仮想回路素子が表す処理内容の実行結果が、実行結果表示手段により表示装置の表示画面上に表示されるため、作業者は、その表示を見ることにより、所望の処理が正しく行われるか否かを検証することができる。

【0031】つまり、このシミュレーション装置によれば、疑似回路図の全体によって表される処理内容のうち、端子選択手段により端子を選択した仮想回路素子から、モニタ対象選択手段により端子を選択した仮想回路素子までの任意の処理内容が、思うように正しく行われるか否かを、プログラムの生成作業を行う前に事前にチェックすることができ、所望のプログラムを作成する際のトータルの作業効率を向上させることができる。

【0032】ところで、前述したプログラム作成装置は、専用の機器として構成するようにしても良いが、次のような機械可読式の媒体を用いれば、汎用のコンピュータシステムを前述したプログラム作成装置として用いることができるようになる。即ち、具体的には請求項9に記載したように、請求項1又は請求項2に記載の素子情報記憶手段を備えたコンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図を解析することで、該疑似回路図により表現されたプログラムを作成するプログラム作成装置として機能させるための処理プログラムを記録した機械読み取り可能な媒体であって、前記記録される処理プログラムは、コンピュータを、前記疑似回路図を読み込む読込手段と、該読込手段により読み込まれた疑似回路図中の

各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、当該疑似回路図により表現されたプログラムの各機能部分として用いられる小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の結合関係を求め、該結合関係に従って前記各小プログラムを結合したプログラムを構築するプログラム構築手段と、して機能させるためのプログラムである機械可読式の媒体を用いれば良い。

【0033】また、前述した描画装置も、専用の機器として構成するようにしても良いが、次のような機械可読式の媒体を用いれば、汎用のコンピュータシステムを前述した描画装置として用いることができるようになる。即ち、具体的には請求項10に記載したように、請求項1又は請求項2に記載の素子情報記憶手段を備えたコンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線した疑似回路図を描画するための描画装置として機能させるための処理プログラムを記録した機械読み取り可能な媒体であって、前記記録される処理プログラムは、コンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を、所定の表示装置の表示画面における所定領域に一覧表示させる一覧表示手段と、前記所定領域に表示された仮想回路素子の中から、作業による操作に従って何れかの仮想回路素子を選択し、該選択した仮想回路素子を、前記表示装置の表示画面において前記所定領域とは異なる領域として設定された編集領域に表示させる素子選択手段と、該素子選択手段により前記編集領域に表示された複数の仮想回路素子間を、作業による操作に従って信号線により該編集領域上で結線する結線手段と、前記編集領域に表示された仮想回路素子及び信号線からなる疑似回路図を所定の記憶媒体に記録させる記録制御手段と、して機能させるためのプログラムである機械可読式の媒体を用いれば良い。

【0034】また更に、前述したシミュレーション装置も、専用の機器として構成するようにしても良いが、次のような機械可読式の媒体を用いれば、汎用のコンピュータシステムを前述したシミュレーション装置として用いることができるようになる。

【0035】即ち、具体的には請求項11に記載したように、請求項1又は請求項2に記載の素子情報記憶手段を備えたコンピュータを、前記素子情報記憶手段に前記小プログラム及び前記端子情報が記憶された複数の仮想回路素子を信号線により結線して描画された疑似回路図によって表現される処理内容を検証するためのシミュレーション装置として機能させるための処理プログラムを記録した機械読み取り可能な媒体であって、前記記録される処理プログラムは、コンピュータを、前記疑似回路

図を読み込む読込手段と、該読込手段により読み込まれた前記疑似回路図を、所定の表示装置の表示画面上に表示させる疑似回路図表示手段と、前記表示された疑似回路図中の何れかの仮想回路素子が有する前記第1又は第3の端子を、作業による操作に従って選択する端子選択手段と、前記表示された疑似回路図中の各仮想回路素子を抽出し、前記素子情報記憶手段を参照することによって、前記各仮想回路素子に対応する小プログラムを特定すると共に、前記疑似回路図中の各仮想回路素子間の結線状態を前記素子情報記憶手段に記憶された端子情報に従って解析することにより、前記特定される各小プログラム同士の実行順序関係を求め、前記特定される各小プログラムの表す処理内容を、前記求めた実行順序関係に従って、前記端子選択手段により前記第1又は第3の端子が選択された仮想回路素子に対応する処理内容から順次実行するプログラム実行手段と、前記表示された疑似回路図中の仮想回路素子のうち、前記プログラム実行手段による実行結果をモニタしたい処理内容に対応する仮想回路素子が有する端子を、作業による操作に従って選択するモニタ対象選択手段と、該モニタ対象選択手段により端子が選択された仮想回路素子が表す処理内容の前記プログラム実行手段による実行結果を、前記表示装置の表示画面上に表示させる実行結果表示手段と、して機能させるためのプログラムである機械可読式の媒体を用いれば良い。

【0036】なお、請求項9～請求項11に記載の機械読み取り可能な媒体としては、フロッピーディスク（FD）、コンパクトディスク（CD-ROM）、磁気テープなどの外部メモリが一般的であるが、ROMやRAMなどの内部メモリであっても良い。

【0037】ところで、請求項1ないし請求項6の何れかに記載のプログラム作成装置において、請求項12に記載のように、プログラム構築手段が、前記疑似回路図により表現されたプログラムの各機能部分として用いられる前記各小プログラムを結合する際に、データのやり取りを行う2つの小プログラム間にて夫々扱われるデータの形態が異なっているか否かを判定し、前記両データの形態が異なっている場合には、前記両データの形態を一致させるためのデータ形態変換用プログラムを作成して前記構築するプログラムに加える、というように構成すれば、所望のプログラムを一層簡単に作成することができる。

【0038】つまり、各仮想回路素子に夫々対応する小プログラムで扱われるデータの形態（例えば、データが示す値の単位や、一度に取り扱うデータのビット数、或いは、後述する実施例で説明するLSBやOFFSETなど）を、事前に厳しくチェックしておく必要がなく、プログラムを作成する際の作業効率を著しく向上させることができるのである。

【0039】一方、請求項13に記載のように、請求項

1ないし請求項6及び請求項12の何れかに記載のプログラム作成装置を、内燃機関を制御する制御プログラムを作成するための装置として用いれば、この種の複雑な制御プログラムを簡単に作成することができるようになる。

【0040】

【発明の実施の形態】以下、本発明が適用された実施例について図面を用いて説明する。尚、本発明の実施の形態は、下記の実施例に何ら限定されることなく、本発明の技術的範囲に属する限り、種々の形態を採り得ることは言うまでもない。

【0041】まず図1は、実施例の制御プログラム開発支援装置1の構成を表す構成図である。本実施例の制御プログラム開発支援装置1は、図2に示すような回路図（疑似回路図）の形で制御システムの仕様（換言すれば、作成しようとするプログラムの処理内容を示すソフトウェアの仕様）を記述して、その回路図からC言語によるプログラムを自動作成し、或いは、上記回路図によって表現される処理内容をシミュレーションして事前に検証するものであり、図1に示す如く、汎用のコンピュータ本体3と、該コンピュータ本体3に対して入力を行うためのキーボード5及びマウス7からなる入力装置と、コンピュータ本体3からの指令に応じて、その表示画面H上に画像を表示するCRT或いは液晶などからなる表示装置9と、データ記憶用のハードディスクユニット11と、CD-ROMドライバ13と、を備えた周知のコンピュータシステムによって構成されている。

【0042】そして、この制御プログラム開発支援装置（以下、単に、支援装置という）1は、上記回路図を作成及び編集するための回路図作成用プログラム（以下、回路エディタソフトという）と、この回路エディタソフトの実行により作成された回路図からC言語によるプログラムを自動作成するためのプログラム作成用プログラム（以下、プログラムジェネレータソフトという）と、上記回路エディタソフトの実行により作成された回路図が表す処理内容を検証するためのシミュレーション用プログラム（以下、シミュレータソフトという）と、上記回路エディタソフトの実行により上記回路図を作成及び編集する際に用いるデータ群（以下、回路素子データベースという）とが予め記録されたCD-ROM15を、CD-ROMドライバ13に装着し、このCD-ROM15の記録内容をハードディスクユニット11にインストールすることにより、コンピュータ本体3を当該支援装置1の主要部として機能させている。

【0043】即ち、コンピュータ本体3が回路エディタソフトを実行することにより、当該支援装置1は、上記回路図を表示装置9の表示画面H上で作成及び編集するための回路エディタとして機能し、コンピュータ本体3がプログラムジェネレータソフトを実行することにより、当該支援装置1は、上記作成された回路図からプロ

グラムを自動作成するためのプログラムジェネレータとして機能し、コンピュータ本体3がシミュレータソフトを実行することにより、当該支援装置1は、上記作成された回路図が表す処理内容を検証するためのシミュレータとして機能する。

【0044】尚、上記各ソフト及び回路素子データベースは、1枚のCD-ROM15ではなく、複数のCD-ROMに夫々記録されていても良い。また、CD-ROMドライバ13に代えて、フロッピーディスクドライバを備えたコンピュータシステムの場合には、上記各ソフト及び回路素子データベースが記録されたフロッピーディスクを上記フロッピーディスクドライバに装着して、それらをハードディスクユニット11にインストールすれば良い。一方、上記各ソフト及び回路素子データベースを、通信によってダウンロードするようにしても良い。

【0045】ここで、コンピュータ本体3が回路エディタソフトを実行して当該支援装置1が回路エディタとして機能することにより作成される回路図（以下、データ回路ともいう）、及び該データ回路を表す回路図データ（以下、回路モデルともいう）について説明する。

【0046】まず、データ回路は、図2(a)にその一例を示すように、四角形のブロックで表示される回路素子（仮想回路素子）と、図2にてE、T、D、Aの各アルファベットで始まる符号によって表される端子と、その端子間を接続する信号線とから構成される。そして、データ回路は、端子を持つ最も外側の回路枠の中に、必要な回路素子を配置し、それら各回路素子の端子相互間あるいは回路枠上の端子との間を、信号線で結線することにより構成され、このように構成した回路図も、1つの回路素子となる。

【0047】つまり、回路素子には、プリミティブな回路素子であって、その動作内容などが上記回路素子データベース中で予め定義された基本素子と、他の回路素子を組み合わせ（結線して）作られている組合せ素子との2種類があり、組合せ素子を階層的に作成することで、ソフトウェアの仕様を各回路素子の階層構造で表現できるようになっている。

【0048】例えば、図2(a)において、2つの回路素子KS1、KS2は基本素子であるが、他の回路素子KS3は、図2(b)に示す如く、基本素子である2つの回路素子KS4、KS5を組み合わせ（結線した）組合せ素子である。次に、このようなデータ回路において、回路素子の四辺には夫々別々の意味が持たされており、各辺に配置される端子の役割を次のように定義付けしている。

【0049】即ち、回路素子の上辺に配置される端子は、その回路素子を駆動するトリガ信号を外部から受信し、それに対応する動作の完了を知らせる完了信号を発信するためのイベント端子（以下、E端子という）とし

て定義付けされており、回路素子の下辺に配置される端子は、他の回路素子を駆動するトリガ信号を発信し、該他の回路素子から動作の完了を知らせる完了信号を受信するためのトリガ端子（以下、T端子という）として定義付けされている。また、回路素子の右辺に配置される端子は、内部のデータに対するアクセス信号を外部から受信し、それに対応するデータを応答するデータ信号を発信するためのデータ端子（以下、D端子という）として定義付けされており、回路素子の左辺に配置される端子は、外部のデータをアクセスするアクセス信号を発信し、それに応答するデータ信号を受信するためのアクセス端子（以下、A端子という）として定義付けされている。

【0050】よって、E端子は他の回路素子のT端子に接続され、T端子は他の回路素子のE端子に接続され、D端子は他の回路素子のA端子に接続され、A端子は他の回路素子のD端子に接続される。また、後述する定義素子と要素素子との間では、定義素子のE端子と要素素子のE端子、定義素子のT端子と要素素子のT端子、定義素子のD端子と要素素子のD端子、定義素子のA端子と要素素子のA端子、というように同じ種類の端子同士が接続される。尚、図2において、符号E1～E5は、E端子を表しており、T1～T3は、T端子を表しており、D1～D5は、D端子を表しており、符号A1～A13は、A端子を表している。

【0051】そして、データ回路では、このような端子の機能に対応して、上記端子間を接続する信号線を2種類に区別している。即ち、T端子とE端子を接続する信号線は、前述したトリガ信号を往路信号として流すと共に前述した完了信号を復路信号として流す、トリガ線として定義付けされており、A端子とD端子を接続する信号線は、前述したアクセス信号を往路信号として流すと共に前述したデータ信号を復路信号として流す、データ線として定義付けされている。

【0052】ところで、図2に例示した回路図で用いられている各基本素子KS1、KS2、KS4、KS5の動作内容（機能）は、夫々、次のようである。基本素子KS1は、ストア素子であり、E端子E3にトリガ信号が来ると、A端子A5からアクセス信号を出力し、それに応答するデータ信号が来ると、そのデータ値を内部にストアし、次にT端子T3からトリガ信号を出力し、それに対応する完了信号が来ると、E端子E3から完了信号を返す。また、E端子E4にトリガ信号が来ると、内部のストア値を0にして、E端子E4から完了信号を返す。また更に、D端子D2にアクセス信号が来ると、内部のストア値をD端子D2から返す。

【0053】基本素子KS2は、スイッチ（SW）素子であり、E端子E5にトリガ信号が来ると、A端子A6からアクセス信号を出力し、それに応答するデータ信号が来ると、そのデータ値が0ならば、そのままE端子E

5から完了信号を返し、逆に、上記データ値が0以外ならば、T端子T2からトリガ信号を出して、それに対応する完了信号が来ると、E端子E5から完了信号を返す。

【0054】基本素子KS4は、乗算素子であり、D端子D4にアクセス信号が来ると、A端子A10とA端子A11から夫々アクセス信号を出力し、それらに夫々応答する2つのデータ信号が来ると、その2つのデータ値の積をD端子D4から返す。基本素子KS5は、加算素子であり、D端子D5にアクセス信号が来ると、A端子A12とA端子A13から夫々アクセス信号を出力し、それらに夫々応答する2つのデータ信号が来ると、その2つのデータ値の和をD端子D5から返す。

【0055】これにより、図2（a）に例示した回路は、次のような動作をする。まず、E端子E1にトリガ信号が来ると、A端子A1、A2、A3から夫々アクセス信号を出力して、それらに夫々応答するデータを取得する。そして、 $A1 * (A2 + A3)$ なる計算をし、その結果をストアする。尚「*」は乗算を表している。その後、A端子A4からアクセス信号を出力して、取得したデータが0ならば、E端子E1から完了信号を返し、逆に、取得したデータが0以外ならば、T端子T1からトリガ信号を出力して、それに対応する完了信号が来ると、E端子E1から完了信号を返す。

【0056】また、E端子E2にトリガ信号が来ると、内部のストア値を0にして、E端子E2から完了信号を返す。また更に、D端子D1にアクセス信号が来ると、ストア値をD端子D1から返す。一方、本実施例において、当該支援装置1が回路エディタとして機能し、上記のようなデータ回路を作成及び編集する際には、各回路素子のA端子及びD端子について、その端子に流れるデータ信号のデータ属性（換言すれば、その端子を流れるデータが持つべきデータ属性）を指定できるようになっている。

【0057】データ属性は、データの内容を示す内容属性（データの単位）と、データ値のコンピュータ上での表現形式である内部表現形式とからなり、内部表現形式には、浮動小数点形式と、LSB・OFFSET形式との2つがある。そして、浮動小数点形式では、データ型だけを指定するようになっており、LSB・OFFSET形式では、データ型、LSB、及びOFFSETの夫々を指定するようになっている。尚、データ型とは、一度に取り扱うデータが何ビット（例えば、2バイト或いは4バイト）のものであるかを示し、LSBとは、取り扱うデータの最下位ビットの値を示し、OFFSETとは、取り扱うデータの全ビットが0であった場合の値を示すものである。

【0058】そして、後述するように、当該支援装置1がシミュレータとして或いはプログラムジェネレータとして機能した際には、信号線により結線されたA端子と

D端子の夫々に指定されているデータ属性が比較され、両端子のデータ属性が異なる場合には、データ属性の変換が自動的に行われるようになっている。

【0059】次に、このようなデータ回路を表す回路図データであって、回路エディタによって作成及び編集される回路モデルは、図3に示す各オブジェクト（データのかたまり）を組み合わせる構成されている。尚、以下の説明において、回路エディタによって既に作成・登録されているか或いは現在作成・編集途中の組合せ素子であって、外側の回路枠に相当する回路素子を、定義素子と言い、その定義素子の構成要素となっている回路素子を、要素素子と言う。よって、要素素子は、初めから回路素子データベースとして用意された基本素子であるか、或いは、既に作成されてハードディスクユニット11内の所定記憶領域に追加の回路素子データベースとして登録された他の定義素子である。また、以下の説明において、定義素子の端子を、定義端子と言い、要素素子の端子を、要素端子と言う。

【0060】まず、回路モデルは、図3(a)～(f)の夫々に示すように、定義素子のオブジェクトと、要素素子のオブジェクトと、端子のオブジェクトと、信号線のオブジェクトと、単位のオブジェクトと、基本素子のオブジェクトとから階層的に構成される。尚、以下では、定義素子のオブジェクトを「定義素子」、要素素子のオブジェクトを「要素素子」、端子のオブジェクトを「端子」、といった具合に、データのかたまりを指す各オブジェクトを「」内の名称で示す。

【0061】即ち、図3(a)に示すように、「定義素子」には、当該定義素子を示す「定義素子名」と、当該定義素子の表示画面H上における表示位置を示す「回路枠座標」と、当該定義素子に設けられた端子（定義端子）の個数（ $=n$ ）を示す「定義端子の個数」と、その各定義端子 x （ x は1～ n の何れか）を夫々表すオブジェクトである各「端子」の格納先（記憶先）を示す「端子 x へのポインタ」と、当該定義素子に含まれる要素素子の個数（ $=m$ ）を示す「要素素子の個数」と、その各要素素子 x （ x は1～ m の何れか）を夫々表すオブジェクトである各「要素素子」の格納先を示す「要素素子 x へのポインタ」と、当該定義素子に含まれる信号線の個数（ $=k$ ）を示す「信号線の個数」と、その各信号線 x （ x は1～ k の何れか）を夫々表すオブジェクトである各「信号線」の格納先を示す「信号線 x へのポインタ」と、からなる情報が含まれている。

【0062】また、図3(b)に示すように、「要素素子」には、当該要素素子を示す「要素素子名」と、当該要素素子の定義素子内での表示配置位置を示す「配置座標」と、当該要素素子を表すオブジェクトである「定義素子」或いは「基本素子」の格納先を示す「定義素子／基本素子へのポインタ」と、当該要素素子に設けられた端子（要素端子）の個数（ $=n$ ）を示す「要素端子の個

数」と、その各要素端子 x （ x は1～ n の何れか）を夫々表すオブジェクトである各「端子」の格納先（記憶先）を示す「端子 x へのポインタ」と、からなる情報が含まれている。

【0063】また更に、図3(c)に示すように、「端子」には、当該端子を示す「端子名」と、当該端子の種類がE端子、T端子、D端子、及びA端子のうちの何れであるかを示す「端子種別」と、当該端子が配置される回路素子での当該端子の表示配置位置を示す「配置座標」と、当該端子が接続されている相手先の端子（接続先端子）を表すオブジェクトである「端子」の格納先を示す「接続先端子へのポインタ」と、当該端子が接続されている信号線を表すオブジェクトである「信号線」の格納先を示す「信号線へのポインタ」と、当該端子に流れるデータ信号の単位を表すオブジェクトである「単位」の格納先を示す「単位へのポインタ」と、当該端子に流れるデータ信号のデータ型、LSB、及びOFFSETを夫々示す、「データ型」と、「LSB」と、「OFFSET」と、からなる情報が含まれている。

【0064】尚、「単位へのポインタ」、「データ型」、「LSB」、及び「OFFSET」は、当該端子がA端子又はD端子の場合にのみ情報が設定される。また、当該端子がA端子又はD端子であっても、取り扱うデータの内部表現形式が浮動小数点形式である場合には、「LSB」及び「OFFSET」としては、情報が設定されない。

【0065】そして、図3(d)に示すように、「信号線」には、当該信号線の一端に接続されている端子を表すオブジェクトである「端子」の格納先を示す「端子1へのポインタ」と、当該信号線の他端に接続されている端子を表すオブジェクトである「端子」の格納先を示す「端子2へのポインタ」と、当該信号線の表示画面H上での形状（詳しくは、当該信号線の表示画面H上での屈曲点の座標列）を示す「形状」と、からなる情報が含まれている。

【0066】また、図3(e)に示すように、「単位」には、当該単位を示す「単位名」と、当該単位から変換が可能な他の単位の種類数（ $=n$ ）を示す「変換可能な単位の数」と、その変換可能な単位 x （ x は1～ n の何れか）を示す「単位 x の単位名」と、当該単位から単位 x へ変換するための変換式を示す「単位 x への変換式」と、からなる情報が含まれている。

【0067】そして、図3(f)に示すように、「基本素子」には、当該基本素子を示す「基本素子名」と、当該基本素子に設けられた端子の個数（ $=n$ ）を示す「端子の個数」と、その各端子 x （ x は1～ n の何れか）を夫々表すオブジェクトである各「端子」の格納先を示す「端子 x へのポインタ」と、当該基本素子のシミュレーション時の動作を表す「シミュレーション動作」と、当該基本素子に対応した処理内容を表す「生成プログラ

ム' と、からなる情報が含まれている。

【0068】上記' シミュレーション動作' は、E 端子
或いはD 端子に信号を受けた場合の当該基本素子の動作
内容を示すものであり、当該支援装置 1 がシミュレータ
として機能した際に、コンピュータ本体 3 が直接実行可
能なコマンド列として記載されている。また、' 生成プ
ログラム' は、当該基本素子に対応した処理内容を表す
プログラムをC 言語によるテキストコードで記載・格納
したものであり、当該支援装置 1 がプログラムジェネレ
ータとして機能した際に抽出される。

【0069】ここで、このような「基本素子」と、その
「基本素子」内の' 端子 x へのポインタ' によって示さ
れる「端子」とは、複数種類の基本素子について、「単
位」と共にCD-ROM 15 内に回路素子データベース
として予め記録されており、この回路素子データベー
スが、前述した如く最初にハードディスクユニット 11 の
所定記憶領域にインストールされる。尚、この時点で、
「基本素子」に対応する「端子」内の情報としては、'
端子名'、' 端子種別'、及び' 配置座標' だけが設定
されている。

【0070】そして、その後、後述する回路エディタに
よって基本素子を組み合わせる組合せ素子が定義素
子として登録・記憶され、或いは更に、それら組合せ素
子や基本素子を組み合わせる組合せ素子が定義素子
として登録・記憶されることにより、初期の回路素子デ
ータベースが、上記の如き組合せ素子群及び基本素子群
からなる回路素子データベースとしてボリュームアップ
される。尚、以下の説明において、このような回路素子
データベースを構成する各回路素子を、登録回路ともい
う。

【0071】また、「定義素子」は、その内部に設定さ
れたポインタが示す各オブジェクトと一緒にファイル化
され、そのファイル（以下、定義素子ファイルという）
に固有のファイル名（以下、定義素子ファイル名とい
う）が付けられた状態でハードディスクユニット 11 内
に記憶される。即ち、「定義素子」は、' 端子 x へのポ
インタ' が示す「端子」、' 要素素子 x へのポインタ'
が示す「要素素子」、及び' 信号線 x へのポインタ' が
示す「信号線」と一緒にファイル化されて記憶され、こ
れにより、その定義素子である回路素子が登録回路とし
て登録される。

【0072】次に、回路エディタについて説明する。使
用者が、キーボード 5 又はマウス 7 を操作して回路エ
ディタソフトを起動すると、当該支援装置 1 は、回路エ
ディタとして動作を開始する。尚、回路エディタは、表示
画面 H に表示された定義素子や端子等の部材をマウス 7
により選択すると、その選択された部材に対応するデー
タのかたまりとしてのオブジェクトを処理の対象とす
る、所謂オブジェクト指向のものであり、以下の説明で
は、上記選択される表示部材もオブジェクトと呼ぶ場合

がある。

【0073】この回路エディタでは、図 4 に示すよう
に、表示装置 9 の表示画面 H を、メニュー画面 17 と、
登録回路一覧表示画面 19 と、回路編集画面 21 との、
少なくとも 3 つの画面に分割する、所謂マルチウインド
ウの表示を行う。そして、メニュー画面 17 には、前述
した定義素子ファイルに対するオープン／セーブ等のフ
ァイル操作のための指示内容や、オブジェクトに対する
処理を指示するための指示内容が表示され、また、登録
回路一覧表示画面 19 には、現在登録されている登録回
路（即ち、既に登録されている回路素子）のアイコン
（シンボル）やファイル名などが表示される。

【0074】そして、使用者は、キーボード 5 やマウス
7 を操作することで、回路編集画面 21 上に回路枠を表
示させると共に、その回路枠に端子を設け、更に、登録
回路一覧表示画面 19 に表示された所望の回路素子（ア
イコン）を、マウス 7 の操作によりドラック及びドロッ
プすることで、回路編集画面 21 上の上記回路枠内に配
置することができる。そして更に、回路編集画面 21 上
に表示された回路枠の端子と、その回路枠内の各回路素
子の端子との間、或いは、回路枠内の各回路素子の端子
間を、マウス 7 の操作により回路編集画面 21 上で結線
することができるようになっている。

【0075】そして、回路エディタでは、このように回
路編集画面 21 上で描画される回路図に対応して、図 3
に示した各オブジェクトを新規作成（メモリに確保）し
たり、削除（メモリを開放）したり、データ値を設定す
ることによって、回路モデルを変更し処理を進めてい
く。つまり、キーボード 5 やマウス 7 を用いた使用者か
らの入力を受け、その入力シーケンスにより何れの処理
を行うかを判定し、その処理に従って回路モデルを変更
しつつ、その結果（即ち、回路図）を回路編集画面 21
上に表示するのである。

【0076】ここで、回路エディタの操作は、概ね、キ
ーボード 5 又はマウス 7 により、オブジェクトを選択
し、その選択したオブジェクトに対する処理を指定する
（メニュー選択する）という手順で行われる。このた
め、図 5 に示すように、回路エディタの中核を成す制御
部は、選択されているオブジェクトへのポインタを保持
する選択リスト（複数のポインタを保持可能）と、一番
最近に選択されたオブジェクトへのポインタを保持する
操作対象とを備えている。

【0077】尚、図 5 は、当該支援装置 1 が回路エディ
タとして作動する場合の各部の関係を概念的に表したブ
ロック図であり、この図 5 における制御部は、回路エ
ディタソフトによってなされる処理の中核部分を表して
いる。そして、上記選択リストと操作対象は、夫々、メモ
リ上に確保された記憶領域である。また、図 5 における
入力部は、キーボード 5 及びマウス 7 と、これらからの
入力を受ける処理の部分を表しており、表示部は、表示

装置 9 と、この表示装置 9 に画像を表示させる処理の部分を表している。

【0078】そこで、このような回路エディタ（回路エディタソフト）の基本的な処理について、図 6～図 19 を用いて説明する。尚、図 6 は、回路エディタの処理全体を表すフローチャートであり、図 7～図 19 は、図 6 の処理中で実行される各処理を表すフローチャートである。

【0079】まず、図 6 に示すように、キーボード 5 及びマウス 7 からの入力を受け（S100、S110）、その入力内容に応じて、何れの処理を行うかを判定する（S120）。そして、その判定結果により、以下の何れかの処理を行う。まず、〈終了〉を指示する入力であれば、当該回路エディタを終了する（S130）。

【0080】一方、既に登録されている定義素子と呼び出すための〈定義素子オープン〉を指示する入力であれば（S140）、図 7 に示す定義素子オープンの処理を行う。即ち、まず、オープンしたいファイル名の入力を使用者に対しダイアログボックスによって促し、定義素子ファイル名を取得する（S141）。そして、このようにして指定された定義素子ファイルをオープンし（S142）、そのファイル内の全オブジェクトを読み込んで（S143）、その読み込んだオブジェクトを表示する（S144）。これにより、その定義素子ファイルに格納されていた定義素子の回路図が、回路編集画面 21 上に表示され、その後、この処理を抜けて図 6 の S100 へ戻る。

【0081】尚、ダイアログボックスとは、表示画面 H の所定位置にファイル名等の所定事項の入力を求めるメッセージを表示する、といった周知のものである。また、作成・編集した定義素子を登録回路として保存するための〈定義素子セーブ〉を指示する入力であれば（S150）、図 8 に示す定義素子セーブの処理を行う。即ち、まず、ファイル名の入力を使用者に対しダイアログボックスによって促し、定義素子ファイル名を取得する（S151）。そして、このように指定された定義素子ファイルをオープンし（S152）、オブジェクト（即ち、編集中的「定義素子」及びそれに含まれる「端子」、「要素素子」、「信号線」）を書き込む（S153）。これにより、回路編集画面 21 上に表示されている定義素子の回路モデルが、指定された定義素子ファイルに格納され、その後、この処理を抜けて図 6 の S100 へ戻る。

【0082】また、定義素子を新規に作成するための〈新定義素子作成〉を指示する入力であれば（S160）、図 9 に示す新定義素子作成の処理を行う。即ち、まず、全ての情報が空である新しい「定義素子」（以下、 $\alpha 1$ と呼ぶ）を作成する（S161）。そして、定義素子名の入力を使用者に対しダイアログボックスによって促し、定義素子名を取得して、その定義素子名を α

1 の「定義素子名」に設定する（S162）。そして更に、 $\alpha 1$ の「回路枠座標」に予め定められた初期値を設定し（S163）、その「回路枠座標」によって指定される回路編集画面 21 上の所定位置に、新規に作成する定義素子の回路枠を表示する（S164）。そして、その後、この処理を抜けて図 6 の S100 へ戻る。

【0083】一方、使用者がマウス 7 を操作して回路編集画面 21 に表示された何れかのオブジェクト（即ち、定義素子、要素素子、端子、信号線の何れか）を指定した場合の〈オブジェクト選択〉であれば（S170）、図 10 に示すオブジェクト選択の処理を行う。即ち、まず、マウス 7 で指定された座標にあるオブジェクト（以下、 $\alpha 2$ と呼ぶ）へのポインタを、選択リストの先頭に加える（S171）。そして、 $\alpha 2$ へのポインタを操作対象に設定し、その後、この処理を抜けて図 6 の S100 へ戻る。

【0084】また、上記の如く行われたオブジェクトの選択を解除するための〈オブジェクト選択解除〉を指示する入力であれば（S180）、図 11 に示すオブジェクト選択解除の処理を行う。即ち、選択リスト及び操作対象を空にし（S181）、その後、この処理を抜けて図 6 の S100 へ戻る。

【0085】また、回路編集画面 21 に現在表示されている回路図（定義素子の回路図）の表示位置を変更するための〈回路枠座標変更〉を指示する入力であれば（S190）、図 12 に示す回路枠座標変更の処理を行う。即ち、まず、操作対象が定義素子か否か（つまり、前述したオブジェクト選択の処理で一番最近に操作対象へ設定されたポインタが「定義素子」へのポインタであるか否か）を判定し（S191）、定義素子でなければ、エラーとして、この処理を終了する（S191: NO）。これに対し、操作対象が定義素子であれば（S191: YES）、新たな回路枠座標の入力を使用者に対しダイアログボックスによって促し、回路枠座標を取得して、その取得した回路枠座標を現在表示されている回路図に対応する「定義素子」の「回路枠座標」に設定する（S192）。そして、その「回路枠座標」により指定される回路編集画面 21 上の位置に、現在編集中の回路素子（定義素子）の回路枠を表示し直し（S193）、その後、この処理を抜けて図 6 の S100 へ戻る。尚、この回路枠座標の変更により、回路枠内の要素素子や信号線等の全ての部材も、その表示位置が変更される。

【0086】一方、回路編集画面 21 に現在表示されている定義素子内に新たな要素素子を追加するための〈要素素子追加〉を指示する入力であれば（S200）、図 13 に示す要素素子追加の処理を行う。即ち、まず、操作対象が既に登録されている定義素子又は基本素子であるか否かを判定し（S201）、そうでなければ、エラーとして、この処理を終了する（S201: NO）。

【0087】これに対して、操作対象が既に登録されて

いる定義素子又は基本素子（以下、 $\alpha 3$ と呼ぶ）であれば（S201: YES）、全ての情報が空である新しい「要素素子」（以下、 $\alpha 4$ と呼ぶ）を作成する（S202）。そして、 $\alpha 3$ へのポインタを、 $\alpha 4$ の「定義素子／基本素子へのポインタ」に設定し（S203）、次に、要素素子名の入力を使用者に対しダイアログボックスによって促し、要素素子名を取得して、その要素素子名を $\alpha 4$ の「要素素子名」に設定する（S204）。そして更に、マウスで現在指定されている座標を、 $\alpha 4$ の「配置座標」に設定し（S205）、また、現在編集している「定義素子」の「要素素子へのポインタ」に $\alpha 4$ へのポインタを追加する。そして、S205で設定した座標に基づき、回路編集画面21上にて、現在編集中の回路素子（定義素子）の回路枠内に、 $\alpha 4$ を示すシンボル（四角形のブロック）を追加して表示し（S207）、その後、この処理を抜けて図6のS100へ戻る。

【0088】また、回路編集画面21に現在表示されている定義素子内から要素素子を削除するための〈要素素子削除〉を指示する入力であれば（S210）、図14に示す要素素子削除の処理を行う。即ち、まず、操作対象が要素素子か否かを判定し（S211）、操作対象が要素素子でなければ、エラーとして、この処理を終了する（S211: NO）。

【0089】これに対して、操作対象が要素素子（以下、 $\alpha 5$ と呼ぶ）であれば（S211: YES）、 $\alpha 5$ が含んでいる全ての「端子」（つまり、要素端子のオブジェクトであり、以下、 $\alpha 6$ と呼ぶ）について、次の処理を繰り返す。まず、 $\alpha 6$ の「信号線へのポインタ」が示す「信号線」（以下、 $\alpha 7$ と呼ぶ）を削除し、更に、 $\alpha 5$ を持つ「定義素子」（以下、 $\alpha 8$ と呼ぶ）の「信号線へのポインタ」から、 $\alpha 7$ に対応する情報を消去する（S213）。そして、 $\alpha 6$ の「接続先端子へのポインタ」が示す「端子」の「接続先端子へのポインタ」を空にし（S214）、 $\alpha 6$ を削除する（S215）。

【0090】そして更に、 $\alpha 5$ が含んでいる全ての「端子」（つまり、要素端子）について、削除が終了すると（S212: YES）、 $\alpha 8$ の「要素素子へのポインタ」から、 $\alpha 5$ に対応する情報を削除し（S216）、次いで $\alpha 5$ を削除する（S217）。その後、回路編集画面21上にて現在編集中の回路素子（定義素子）の回路枠内から、 $\alpha 5$ を示すシンボルを消し（S218）、その後、この処理を抜けて図6のS100へ戻る。

【0091】一方、回路編集画面21に現在表示されている定義素子又は要素素子に端子を追加するための〈端子追加〉を指示する入力であれば（S220）、図15に示す端子追加の処理を行う。即ち、まず、操作対象が定義素子又は要素素子であるか否かを判定し（S221）、そうでなければ、エラーとして、この処理を終了する（S221: NO）。

【0092】これに対して、操作対象が定義素子又は要素素子（以下、 $\alpha 9$ と呼ぶ）であれば（S221: YES）、全ての情報が空である新しい「端子」（以下、 $\alpha 10$ と呼ぶ）を作成する（S222）。そして、端子名の入力を使用者に対しダイアログボックスによって促し、端子名を取得して、その端子名を $\alpha 10$ の「端子名」に設定し（S223）、更に、端子種別の入力を使用者に対しダイアログボックスによって促し、端子種別を取得して、その端子種別を $\alpha 10$ の「端子種別」に設定する（S224）。

【0093】そして更に、 $\alpha 9$ の「端子へのポインタ」に $\alpha 10$ へのポインタを追加し（S225）、また、 $\alpha 9$ の座標と $\alpha 10$ の「端子種別」から、今回追加する新たな端子の配置座標を求め、その座標を $\alpha 10$ の「配置座標」に設定する（S226）。そして、S226で設定した座標に基づき、回路編集画面21上にて、端子追加の対象である回路素子（定義素子又は要素素子）の回路枠に、新たな端子を追加して表示し（S227）、その後、この処理を抜けて図6のS100へ戻る。

【0094】尚、S226での配置座標の設定は、 $\alpha 10$ の端子種別に応じて、E端子ならば回路枠の上辺に、T端子ならば回路枠の下辺に、A端子ならば回路枠の左辺に、D端子ならば回路枠の右辺に、といった具合に設定される。また、回路編集画面21に現在表示されている定義素子又は要素素子から端子を削除するための〈端子削除〉を指示する入力であれば（S230）、図16に示す端子削除の処理を行う。

【0095】即ち、まず、操作対象が端子か否かを判定し（S231）、操作対象が端子でなければ、エラーとして、この処理を終了する（S231: NO）。これに対して、操作対象が端子（以下、 $\alpha 11$ と呼ぶ）であれば（S231: YES）、 $\alpha 11$ の「信号線へのポインタ」が示す「信号線」（以下、 $\alpha 12$ と呼ぶ）を削除し、更に、 $\alpha 12$ を持つ「定義素子」の「信号線へのポインタ」から、 $\alpha 12$ に対応する情報を消去する（S232）。

【0096】そして、 $\alpha 11$ の「接続先端子へのポインタ」が示す「端子」の「接続先端子へのポインタ」を空にし（S233）、更に、 $\alpha 11$ が定義端子の場合には、 $\alpha 11$ を持つ「定義素子」の「端子へのポインタ」から、 $\alpha 11$ に対応する情報を消去し、また、 $\alpha 11$ が要素端子の場合は、 $\alpha 11$ を持つ「要素素子」の「端子へのポインタ」から、 $\alpha 11$ に対応する情報を消去する（S234）。次いで、 $\alpha 11$ を削除すると共に（S235）、回路編集画面21に表示していた $\alpha 11$ のシンボル（端子のシンボル）を消し（S236）、その後、この処理を抜けて図6のS100へ戻る。

【0097】一方、回路編集画面21に現在表示されている2つの端子間を信号線で結線するための〈結線〉を指示する入力であれば（S240）、図17に示す結線

の処理を行う。即ち、まず、操作対象（つまり、最新に選択されたオブジェクト）と選択リストの2番目の要素（つまり、前回に選択されたオブジェクト）とが、両方共に端子であるか否かを判定し（S241）、そうでなければ、エラーとして、この処理を終了する（S241:NO）。

【0098】また、操作対象（以下、 $\alpha 13$ と呼ぶ）と選択リストの2番目の要素（以下、 $\alpha 14$ と呼ぶ）とが、両方共に端子であれば（S241:YES）、つまり、オブジェクトとして、2回連続して端子が選択されている場合には、 $\alpha 13$ と $\alpha 14$ の「端子種別」を調べる。そして、接続できる組み合わせでなければ、エラーとして、この処理を終了する（S242:NO）。尚、接続できる組み合わせは、2つの端子が定義端子同士或いは要素端子同士の場合には、E端子とT端子、又は、D端子とA端子であり、2つの端子が定義端子と要素端子である場合には、E端子とE端子、T端子とT端子、といった具合に同じ種別の端子である。

【0099】これに対して、 $\alpha 13$ と $\alpha 14$ の「端子種別」が、接続できる組み合わせであれば（S242:YES）、全ての情報が空である新しい「信号線」（以下、 $\alpha 15$ と呼ぶ）を作成する（S243）。そして、 $\alpha 15$ の「端子1へのポインタ」に、 $\alpha 13$ へのポインタを設定し、更に、 $\alpha 15$ の「端子2へのポインタ」に、 $\alpha 14$ へのポインタを設定する（S244）。また、 $\alpha 13$ と $\alpha 14$ の「信号線へのポインタ」に、夫々、 $\alpha 15$ へのポインタを設定し、更に、現在編集中的「定義素子」の「信号線へのポインタ」に、 $\alpha 15$ へのポインタを追加する（S245）。

【0100】そして更に、 $\alpha 13$ の「接続先端子へのポインタ」に、 $\alpha 14$ へのポインタを設定すると共に、 $\alpha 14$ の「接続先端子へのポインタ」に、 $\alpha 13$ へのポインタを設定し（S246）、また、 $\alpha 13$ と $\alpha 14$ の「配置座標」から、今回追加する信号線の表示画面H上での形状を求め、その求めた形状を、 $\alpha 15$ の「形状」に設定する（S247）。

【0101】そして、S247で設定した形状に基づき、回路編集画面21上にて、定義素子の回路枠に、新たな信号線を追加して表示し（S248）、その後、この処理を抜けて図6のS100へ戻る。これにより、オブジェクトとして選択された2つの端子間が、回路編集画面21上で信号線により自動的に結線される。

【0102】一方、回路編集画面21に現在表示されている信号線を削除するための〈信号線削除〉を指示する入力であれば（S250）、図18に示す信号線削除の処理を行う。即ち、まず、操作対象が信号線であるか否かを判定し（S251）、操作対象が信号線でなければ、エラーとして、この処理を終了する（S251:NO）。

【0103】これに対して、操作対象が信号線（以下、

$\alpha 16$ と呼ぶ）であれば（S251:YES）、 $\alpha 16$ の「端子1へのポインタ」が示す「端子」（以下、 $\alpha 17$ と呼ぶ）と、 $\alpha 16$ の「端子2へのポインタ」が示す「端子」（以下、 $\alpha 18$ と呼ぶ）との、「信号線へのポインタ」を空にし、更に、「定義素子」の「信号線へのポインタ」から、 $\alpha 16$ に対応する情報を消去する（S252）。

【0104】そして、 $\alpha 17$ と $\alpha 18$ の「接続先端子へのポインタ」を空にして（S253）、 $\alpha 16$ を削除し（S254）、次いで、回路編集画面21に表示していた $\alpha 16$ のシンボル（ライン）を消し（S255）、その後、この処理を抜けて図6のS100へ戻る。

【0105】一方、端子に対するデータ属性（その端子に流れるべきデータのデータ属性）を設定するための〈データ属性設定〉を指示する入力であれば（S260）、図19に示すデータ属性設定の処理を行う。即ち、まず、操作対象が端子であり且つその「端子種別」がD端子又はA端子であるか否かを判定し、そうでなければ、エラーとして、この処理を終了する（S261:NO）。

【0106】これに対して、操作対象が端子（以下、 $\alpha 19$ と呼ぶ）であり且つその「端子種別」がD端子又はA端子であれば（S261:YES）、使用者に対し単位名の入力をダイアログボックスによって促して、単位名を取得し、その取得した単位名を持つ「単位」へのポインタを、 $\alpha 19$ の「単位へのポインタ」に設定する（S262）。

【0107】そして更に、使用者に対しデータ型、LSB、及びOFFSETの入力をダイアログボックスによって夫々促し、その各情報を取得する。そして、その取得したデータ型、LSB、及びOFFSETを、夫々、 $\alpha 19$ の「データ型」、「LSB」、及び「OFFSET」に夫々設定し（S263、S264、S265）、その後、この処理を終了して、図6のS100へ戻る。

【0108】尚、図10に示したオブジェクト選択の処理と、図11に示したオブジェクト選択解除の処理は、図12～図19の各処理で操作対象（つまり、処理対象であるオブジェクト）を特定するために行われるものである。次に、このような回路エディタによって所望の回路図を描画する際の手順の一例について説明する。

【0109】まず、キーボード5又はマウス7を操作して、表示装置9の表示画面Hを、前述したメニュー画面17、登録回路一覧表示画面19、及び回路編集画面21の3つの画面に分割させる。そして、回路図を新規に作成する場合には、マウス7によりメニュー画面17から〈新定義素子作成〉の指示を選択する。すると、図9に示した新定義素子作成の処理（図6のS160）により、回路編集画面21上の所定位置に、新規に作成する定義素子の回路枠が表示される。次いで、この回路枠に端子を設けるのであるが、この場合には、マウス7によ

り、まず、回路編集画面21に表示された回路枠内のどこかの位置を指定し（操作対象を定義素子とし）、次に、メニュー画面17から〈端子追加〉の指示を選択する。すると、図15に示した端子追加の処理（図6のS220）によって、表示画面Hにダイアログボックスが表示されるため、キーボード5により回路枠に設けたい端子の端子種別を入力すれば、その端子種別に応じた回路枠上の位置に、端子が自動的に表示される。

【0110】尚、前述したS226の処理により、端子種別として、E端子を入力すれば、回路枠の上辺に端子が表示され、T端子を入力すれば、回路枠の下辺に端子が表示され、D端子を入力すれば、回路枠の右辺に端子が表示され、A端子を入力すれば、回路枠の左辺に端子が表示される。

【0111】また、このように端子を設ける場合には、当該回路から他の回路に提供する提供データを考えて、必要であれば、回路枠の右辺に、その提供データ毎にデータ信号の提供口となるD端子を設ける。また、当該回路が提供データを求めるのに他の回路から参照したいデータ（外部データ）を考えると、必要であれば、回路枠の左辺に、外部データ毎にデータ信号の参照口となるA端子を設ける。また更に、当該回路の内部状態に変化を起こすイベント（当該回路の動作を起動する契機）を考え、必要であれば、回路枠の上辺に、そのイベント毎にトリガ信号の受信口となるE端子を設ける。そして、当該回路から他の回路の内部状態に変化を起こすイベントを考え、必要であれば、回路枠の下辺に、そのイベント毎にトリガ信号の発信口となるT端子を設ける。

【0112】次に、表示されている回路枠の中に、所望の処理を行うための回路素子を配置するのであるが、この場合には、マウス7により、まず、登録回路一覧表示画面19に表示されている登録回路（既に登録されている回路素子）の中から所望の回路素子を選んで、次に、メニュー画面17から〈要素素子追加〉の指示を選択し、更に、その選んだ回路素子を、マウス7により回路編集画面21上の上記回路枠内に配置する。そして、このような操作を繰り返すことにより、回路枠内の所望の位置に複数の回路素子を配置することができる。尚、この動作は、図13に示した要素素子追加の処理（図6のS200）によって実現され、この時、新たな「要素素子」が作成される。

【0113】そして次に、上記の如く回路枠内に要素素子として配置した各回路素子の端子間、或いは、その要素素子の端子と上記回路枠に設けた端子との間を、信号線によって結線するのであるが、この場合には、マウス7により、まず、回路編集画面21上にて結線したい端子と端子を連続して指定し、次に、メニュー画面17から〈結線〉の指示を選択する。

【0114】すると、図17に示した結線の処理（図6のS240）により、指定した2つの端子間が回路編集

画面21上で信号線により自動的に結線されるため、上記操作を繰り返すことにより、所望の端子間を結線することができる。一方、回路枠内に配置した回路素子（要素素子）を削除する場合には、マウス7により、まず、回路編集画面21上にて削除したい回路素子を指定し、次に、メニュー画面17から〈要素素子削除〉の指示を選択する。すると、図14に示した要素素子削除の処理（図6のS210）により、指定した回路素子が回路編集画面21から消されることとなる。

【0115】また、回路枠に設けた端子を削除する場合には、マウス7により、まず、回路編集画面21上にて削除したい端子を指定し、次に、メニュー画面17から〈端子削除〉の指示を選択する。すると、図16に示した端子削除の処理（図6のS230）により、指定した端子が回路編集画面21から消されることとなる。

【0116】また更に、回路枠内の信号線を削除する場合には、マウス7により、まず、回路編集画面21上にて削除したい信号線を指定し、次に、メニュー画面17から〈信号線削除〉の指示を選択する。すると、図18に示した信号線削除の処理（図6のS250）により、指定した信号線が回路編集画面21から消されることとなる。

【0117】また、回路枠の表示位置を変更する場合には、マウス7により、メニュー画面17から〈回路枠座標変更〉の指示を選択する。すると、図12に示した回路枠座標変更の処理（図6のS190）によって、表示画面Hにダイアログボックスが表示されるため、キーボード5により回路枠の座標を入力すれば、その座標に応じた回路編集画面21上の位置に回路枠が表示し直されることとなる。尚、この場合に、回路枠内の全ての要素も表示し直される。

【0118】一方、回路枠の端子に対するデータ属性を設定する場合には、マウス7により、まず、回路編集画面21上にてデータ属性を設定したい端子（D端子又はA端子）を指定し、次に、メニュー画面17から〈データ属性設定〉の指示を選択する。

【0119】すると、図19に示したデータ属性設定の処理（図6のS260）により、表示画面H上に単位名、データ型、LSB、及びOFFSETを夫々入力するためのダイアログボックスが表示されるため、キーボード5により、その各情報を入力すれば、その入力したデータ属性（単位名、データ型、LSB、及びOFFSET）が上記指定した端子のデータ属性として設定されることとなる。

【0120】以上のような手順によって回路図の回路編集画面21上での描画・編集を行うことができるのであるが、回路エディタでは、図10～図19の処理により、表示されている回路図に対応した回路モデル（回路図データ）を同時に作成・編集している。

【0121】そして、所望の回路図を作ることができた

ならば、次に、この作成した回路図の回路モデルを保存（セーブ）することとなるが、この場合は、マウス7により、メニュー画面17から〈定義素子セーブ〉の指示を選択する。すると、図8に示した定義素子セーブの処理（図6のS150）により、表示画面Hにダイアログボックスが表示されるため、キーボード5によりファイル名（定義素子ファイル名）を入力すれば、作成された回路モデルが上記入力したファイル名と対応付けされてハードディスク11内にセーブされ、新たな登録回路として登録される。

【0122】また、このようにセーブした回路モデルを再度編集したい場合には、マウス7により、メニュー画面17から〈定義素子オープン〉の指示を選択する。すると、図7に示した定義素子オープンの処理（図6のS140）により、表示画面Hにダイアログボックスが表示されるため、キーボード5により編集したい回路モデルのファイル名を入力すれば、その回路モデルを表す回路図が回路編集画面21に表示される。その後、前述した各操作を行うことで、その回路モデルの編集作業を行うことができる。

【0123】一方、当該回路エディタを終了したい場合には、マウス7により、メニュー画面17から〈終了〉の指示を選択すれば良い。このような回路エディタにより描画された回路図に対応する回路モデル（回路図データ）は、図3からも分かるように、「定義素子」を最上位として、その内部に設定されたポイントが示す下位の各オブジェクト（「端子」、「要素素子」、「信号線」）を読み出し、更に、その各オブジェクトの内部に設定されたポイントが示す下位の各オブジェクトを読み出し、更に、その各オブジェクトの内部に設定されたポイントが示す下位の各オブジェクトを読み出す、・・・、といった具合に各オブジェクトを階層的に順次調べていくことにより、最終的に、その回路図（定義素子）の内部に含まれた全ての基本素子及びそれらの端子の信号線による結線状態を全て解析可能なものとなる。

【0124】次に、このような回路エディタにより描画された回路図（即ち、その回路図の回路モデル）が表す処理内容を検証するためのシミュレータについて説明する。尚、前述したように、使用者がキーボード5又はマウス7を操作してシミュレータソフトを起動すると、当該支援装置1はシミュレータとして動作を開始する。

【0125】まず、シミュレータの全体的な概要について説明する。シミュレータは、回路エディタにより作成された回路図内の各回路素子を、その結線状態に応じて疑似的に動作させることにより、その回路図が表す処理内容が期待したものであるか否かを検証するものである。

【0126】そして、シミュレータでは、回路エディタにより作成されて表示画面Hに表示された回路図に対し、シミュレーション用の疑似信号を発生するための回

路素子（以下、信号発生テストツールという）と、回路に流れる信号のモニタを行うための回路素子（以下、モニタテストツールという）との、2種類のシミュレーション専用の回路素子（以下、テストツールという）を接続することによって、上記回路図の動作を検証（シミュレーション）する。

【0127】つまり、信号発生テストツールの端子と、作成された回路図の何れかの端子とを接続することによって、その端子から回路に仮想的な信号（疑似信号）を流すことができ、また、モニタテストツールの端子と、作成された回路図の何れかの端子とを接続することによって、その端子に流れる信号を表示画面Hに表示する（モニタする）ことができる。尚、信号発生テストツールが信号を発生するタイミングやデータ値は、キーボード5又はマウス7の操作により、テストツール毎に設定することができる。

【0128】また、テストツールの端子と回路の端子との接続は、回路エディタの様な信号線ではなく、プローブという特別な表示部材を用いて行う。ここで、テストツールは、図20（a）に示すように、回路モデルの構成要素である「基本素子」と同様のデータ構成（図3（f）参照）で記憶されている。

【0129】即ち、テストツールのオブジェクト（データのかたまりとしてのオブジェクトであって、以下、「テストツール」と記す）は、当該テストツールを示す「テストツール名」と、当該テストツールに設けられた端子の個数（＝n）を示す「端子の個数」と、その各端子x（xは1～nの何れか）を夫々表す「端子」の格納先を示す「端子xへのポイント」と、当該テストツールのシミュレーション時の動作を表す「シミュレーション動作」とからなる。但し、「基本素子」のような「生成プログラム」は設けられていない。

【0130】そして、このような「テストツール」は、「基本素子」と全く同様に、その「テストツール」内の「端子xへのポイント」によって示される「端子」と共に、複数種類のテストツールについてCD-ROM15内に回路素子データベースとして予め記録されており、前述した如く最初にハードディスクユニット11の所定記憶領域にインストールされる。また、前述した回路エディタによって、複数のテストツールを組み合わせた回路を作成し、その回路を組合せ素子の如き組合せテストツールとして登録することもできる。

【0131】一方、上記プローブは、表示画面H上の表示部材としては、例えば矢印状のマークで示されるものであり、内部処理で取り扱うオブジェクト（データのかたまり）としては、図20（b）に示すように、回路モデルの構成要素である「信号線」と同様のデータ構成（図3（d）参照）になっている。

【0132】即ち、プローブのオブジェクト（データのかたまりとしてのオブジェクトであって、以下、「プロ

ープ」と記す)は、当該プローブの一方側に接続されている端子(つまり、テストツール側の端子)を表す「端子」の格納先を示す「端子1へのポインタ」と、当該プローブの他方に接続されている端子(つまり、回路側の端子)を表す「端子」の格納先を示す「端子2へのポインタ」とからなり、「信号線」のような「形状」は設けられていない。

【0133】次に、当該シミュレータによってシミュレーションを実施する前に行う操作、及びそれに対応して実行される処理について説明する。まず、使用者がキーボード5又はマウス7によって所定の入力を行うと、回路エディタの場合と同様に、表示装置9の表示画面Hを、メニュー画面17と、登録回路一覧表示画面19と、回路編集画面21との、少なくとも3つの画面に分割する処理が行われる(図4参照)。但し、この場合に、登録回路一覧表示画面19には、現在登録されているテストツール(即ち、最初にインストールされたテストツールと、それらを組み合わせて登録されたテストツール)のシンボルやファイル名などが表示される。よって以下、シミュレータにおける図4の登録回路一覧表示画面19と同様の画面を、改めて、登録テストツール一覧表示画面と呼ぶ。

【0134】そして、当該シミュレータにおいても、回路エディタの場合と同様に図6～図19に示した処理が行えるようになっており、前述したように、マウス7によってメニュー画面17から〈定義素子オープン〉の指示を選択すれば、図7に示した定義素子オープンの処理により、表示画面Hの所定位置にダイアログボックスが表示される。そこで、キーボード5により、これからシミュレーションしたい回路モデルのファイル名を入力すれば、その回路モデル(回路図データ)が読み込まれると共に、その回路モデルを表す回路図が回路編集画面21に表示される。

【0135】尚、本実施例の支援装置1では、登録回路一覧表示画面19と登録テストツール一覧表示画面との切替だけを行うことも可能であるため、回路エディタの機能によって回路モデルを作成した後に、ファイルセーブをすることなく、そのままこの状態、即ちシミュレーションしたい回路図が回路編集画面21に表示された状態とすることもできる。

【0136】このようにシミュレーションの対象となる回路図が回路編集画面21に表示されたならば、マウス7により、まず、回路編集画面21に表示されている回路内の何れかの端子と、登録テストツール一覧表示画面に表示されているテストツールのうちの何れかの端子とを連続して指定し、次に、メニュー画面17から、表示中の回路内の端子とテストツールの端子とを接続するための〈プローブ接続〉の指示を選択する。

【0137】すると、図17に示した結線の処理とほぼ同様の処理が行われ、指定された両端子の種別が接続可

能な組み合わせであれば、表示画面Hにおいて、その各端子の根元にプローブのマーク(矢印状のマーク)が表示される(図21参照)。そして更に、新たな「プローブ」が作成され、その「プローブ」の「端子1へのポインタ」に、指定されたテストツールの「端子」へのポインタが設定されると共に、該「プローブ」の「端子2へのポインタ」に、指定された回路側の「端子」へのポインタが設定され、このような処理により、回路側の端子とテストツールの端子とのプローブによる接続が行われる。

【0138】そして、上記操作を繰り返すことにより、回路側の複数の端子に、夫々、様々な機能のテストツールを接続することができ、このような作業により、シミュレーションを実施するための準備が終わる。例えば、シミュレーションの対象となる回路が、図2(a)に例示した回路であって、基本素子(ストア素子)KS1のE端子E3にトリガ信号を与えた場合の、組合せ素子KS3の出力データを検証する場合には、図21に示すように、「シミュレーション動作」として所定時間毎にトリガ信号を発信するという動作が設定された信号発生テストツールST1のT端子と、基本素子KS1のE端子E3とをプローブで接続し、「シミュレーション動作」としてD端子にアクセス信号が来ると所定値のデータ信号を返すという動作が設定された3つの信号発生テストツールST2～ST4の各D端子と、組合せ素子KS3の3つのA端子A7～A9とを夫々プローブで接続し、更に、「シミュレーション動作」としてデータ信号の値を表示画面Hの所定位置に表示させるという動作が設定されたモニタテストツールMT1の端子と、組合せ素子KS3のD端子D3とをプローブで接続すれば良い。

【0139】尚、図21において、矢印状のマークは、回路側のプローブを表し、一点鎖線で示す各ブロックは、夫々、登録テストツール一覧表示画面に表示された上記各テストツールST1～ST4、MT1を表している。次に、当該シミュレータにてシミュレーションを実施するために行われる処理の概要について説明する。

【0140】まず、当該支援装置1がシミュレータとして作動する場合の各部の関係を概念的に表すと、図22に示すブロック図のようになる。そして、シミュレータは、テストツールを含む各回路素子の端子が送信した信号を、接続先の端子に伝えると共に、その信号を受信した端子が、基本素子又はテストツールの端子ならば、その基本素子/テストツールに予め定義されている「シミュレーション動作」を起動する、といった2つの処理を繰り返すことにより、シミュレーションを進めていく。

【0141】ここで、信号の伝搬は、発信された信号毎に、図23に示す如きデータのかたまりである信号オブジェクトを生成し、その信号オブジェクトを、メモリ上に確保された記憶領域である信号スタック(図22参照)に登録することによって行う。つまり、信号スタッ

クに登録されている信号オブジェクトは、発信されたが未だ受信されていない疑似的な信号である。

【0142】そして、シミュレータは、信号スタックから信号オブジェクトを取り出して、受信側の端子に渡す。これが信号の受信に相当する。また、信号オブジェクトを渡された端子が基本素子又はテストツールの端子の場合には、その基本素子／テストツールに定義されている「シミュレーション動作」の内容を実行し、それ以外の場合は、渡された信号オブジェクトと接続先端子の情報から新しい信号オブジェクトを生成し、信号スタックに登録する。

【0143】また、シミュレータは、前述した回路エディタによって端子（D端子又はA端子）に設定されたデータ属性と、流れてきたデータのデータ属性とを比較し、両データ属性が異なる場合には、データ属性の自動変換を行う。また、この時、オーバーフローのチェックも行う。

【0144】ところで、信号オブジェクトは、図23に示すように、当該信号の種別を示す「信号種別」と、当該信号を発信した端子（発信端子）を表す「端子」の格納先を示す「発信端子へのポインタ」と、上記発信端子に接続されている端子であって当該信号を受信すべき端子（受信端子）を表す「端子」の格納先を示す「受信端子へのポインタ」と、当該信号がデータ信号である場合にその外部値を示す「データの外部値」と、当該信号がデータ信号である場合にその内部値を示す「データの内部値」と、当該信号がデータ信号である場合にその単位を表す「単位」の格納先を示す「単位へのポインタ」と、当該信号がデータ信号である場合にそのデータ型、LSB、及びOFFSETを夫々示す「データ型」、「LSB」、及び「OFFSET」と、から構成されている。

【0145】尚、上記「信号種別」は、当該信号の発信端子の種別（「端子種別」）に応じて、T端子ならトリガ信号、E端子なら完了信号、A端子ならアクセス信号、D端子ならデータ信号、といった具合に設定される。また、信号オブジェクトの「受信端子へのポインタ」は、当該信号の発信端子に対応する「端子」内の「接続先端子へのポインタ」を参照するか、或いは、その「端子」内の「信号線へのポインタ」により示される「信号線」内の「端子へのポインタ」を参照するか、或いは更に、当該発信端子へのポインタを持つ「プローブ」内の「端子へのポインタ」を参照することにより設定される。つまり、発信端子の接続先端子であって当該信号の受信端子は、「端子」、「信号線」、及び「プローブ」内の情報を参照して求められる。

【0146】一方、データの外部値とは、例えば50kmといった実際の値のことであり、データの内部値とは、例えば5DH（16進数の5D）といったコンピュータ上での表現値のことであり、また、「単位へのポイ

ンタ」、「データ型」、「LSB」、及び「OFFSET」の夫々には、当該信号の発信端子に対応する「端子」内の情報がそのまま設定される（図3（c）参照）。

【0147】次に、このようなシミュレータ（シミュレーションソフト）の基本的な処理について、図24及び図25を用いて説明する。尚、図24は、シミュレータの処理全体を表すフローチャートであり、図25は、図24の処理中で実行されるデータ属性変換処理を表すフローチャートである。

【0148】まず、図24に示すように、回路に接続した信号発生テストツールのうち、キーボード5又はマウス7によって予め指定した信号発生テストツールから、信号（疑似信号）を発生する（S300）。つまり、その信号発生テストツールの端子について信号オブジェクトを生成し、その生成した信号オブジェクトを信号スタックに登録する。

【0149】尚、この最初の場合の信号オブジェクトの「信号種別」は、信号発生テストツールの端子の端子種別で決まる（T端子ならトリガ信号、E端子なら完了信号、A端子ならアクセス信号、D端子ならデータ信号）。また、信号オブジェクトの「発信端子へのポインタ」には、信号発生テストツールの「端子」へのポインタが設定され、「受信端子へのポインタ」には、その「端子」の「接続先端子へのポインタ」と同じ内容が設定される。

【0150】例えば、図21に示した例では、まず、基本素子（ストア素子）KS1のE端子E3に接続された信号発生テストツールST1のT端子から、信号オブジェクトによる疑似信号を発生させる。この場合、信号オブジェクトの「信号種別」はトリガ信号となり、「発信端子へのポインタ」には、信号発生テストツールST1のT端子に対応する「端子」へのポインタが設定され、「受信端子へのポインタ」には、基本素子KS1のE端子E3に対応する「端子」へのポインタが設定されることとなる。

【0151】次に、信号スタックが空になるまで（S310：YES）、以下の処理を繰り返す。まず最初に、信号スタックから、先頭の信号オブジェクト（つまり、信号スタックに最も最近登録された信号オブジェクト）を取り出す（S320）。尚、以下の説明では、この取り出された信号オブジェクトを $\alpha 20$ と呼ぶ。

【0152】そして、 $\alpha 20$ の「信号種別」を判別し（S330）、その「信号種別」がデータ信号の場合には（S330：YES）、 $\alpha 20$ のデータ属性（即ち、単位、データ型、LSB、OFFSET）と、受信端子のデータ属性とが同じか否かをチェックする等の処理を行う後述のデータ属性変換処理を行う（S340）。尚、受信端子のデータ属性は、 $\alpha 20$ の「受信端子へのポインタ」が示す「端子」内の情報から調べる。

【0153】そして、S340のデータ属性変換処理を行うか、或いは、S330で‘信号種別’がデータ信号ではないと判定した場合には(S330:NO)、 $\alpha 20$ の受信端子(以下、 $\alpha 21$ と呼ぶ)が、次の3つの場合の何れに該当するかを判定して(S350)、夫々の場合に依じた処理を行う。

【0154】まず第1に、 $\alpha 21$ が、基本素子又はテストツールの端子である場合には(S360)、その基本素子又はテストツールのシミュレーション動作を起動する(例えば、モニタテストツールの端子に信号が来たときには、その信号の内容を表示画面Hに表示する)。そして、そのシミュレーション動作の起動により、新たに信号を発信する場合(つまり、受信した信号の応答を返したり、他の端子に信号を発信する場合)には、新しく信号オブジェクトを作成して、信号スタックに登録する。尚、シミュレーション動作の起動とは、その基本素子又はテストツールの‘シミュレーション動作’に記載されているコマンド列を実行することである。また、このように新たな信号オブジェクトを作成・登録して、次回にS320からの処理を実行することにより、疑似信号を次の回路素子へと伝搬させる。

【0155】また第2に、 $\alpha 21$ が、基本素子又はテストツールの端子ではなく、且つ、信号線又はプローブの接続された端子である場合には(S370)、 $\alpha 21$ に接続されている全ての信号線又はプローブの接続先端子を受信端子として、新たな信号オブジェクトを作成し、信号スタックに登録する。つまり、ここで作成される信号オブジェクトは、 $\alpha 21$ を発信端子とし、 $\alpha 21$ に接続されている信号線又はプローブの接続先端子を受信端子とし、その他の信号種別などの情報は $\alpha 20$ と同じものとして作成される。そして、このような新たな信号オブジェクトを作成・登録して、次回にS320からの処理を実行することにより、信号線又はプローブ上で疑似信号を下位の回路素子あるいは上位の回路素子へと伝搬させる。

【0156】また第3に、 $\alpha 21$ が、何も接続されていない端子である場合には、 $\alpha 20$ の‘信号種別’を判別し(S380)、その判別結果に応じて、次の3つの何れかの処理を行う。まず、 $\alpha 20$ の‘信号種別’がトリガ信号の場合には(S390)、完了信号の信号オブジェクトを作り、信号スタックに登録する。尚、この場合に作成される信号オブジェクトは、‘信号種別’を完了信号とし、 $\alpha 21$ を発信端子とし、 $\alpha 20$ の発信端子を受信端子として作成される。

【0157】また、 $\alpha 20$ の‘信号種別’がアクセス信号の場合には(S400)、値が0のデータを持つデータ信号の信号オブジェクトを作り、信号スタックに登録する。尚、この場合に作成される信号オブジェクトは、‘信号種別’をデータ信号とし、 $\alpha 21$ を発信端子とし、 $\alpha 20$ の発信端子を受信端子とし、‘データの外部値’及び‘データの内部値’を0とし、データ属性(データ型、LSB等)を予め定められたデフォルト値として作成される。

部値’及び‘データの内部値’を0とし、データ属性(データ型、LSB等)を予め定められたデフォルト値として作成される。

【0158】これに対して、 $\alpha 20$ の‘信号種別’が完了信号又はデータ信号の場合には、何もしない。このようにして上記各S360~S400の何れかの処理が終わると、 $\alpha 20$ (つまり、S320で信号スタックから取り出した信号オブジェクト)を削除する(S410)。そして、使用者による終了の指示があれば、このシミュレータの処理を終了し(S420:YES)、また、終了の指示がなければ(S420:NO)、S310に戻り、信号スタックが空になるまで、S320以降の処理を繰り返す。

【0159】次に、信号スタックから取り出した信号オブジェクト($\alpha 20$)の‘信号種別’がデータ信号である場合に(S330:YES)実行されるデータ属性変換処理(S340)について、図25に沿って説明する。図25に示すように、データ属性変換処理の実行が開始されると、まず、S500にて、信号オブジェクトのデータ属性(即ち、データ信号を発した発信端子のデータ属性であって、流れるデータのデータ属性)と受信端子のデータ属性とが異なっているか否かを判定し、両者のデータ属性が一致していれば(S500:NO)、そのまま当該処理を終了する。尚、受信端子のデータ属性が指定(設定)されていない場合には、両者のデータ属性は一致しているものと見なす。

【0160】これに対して、上記両者のデータ属性が異なっている場合には(S500:YES)、続くS510にて、上記両者の単位が相違しているか否かを判定する。そして、単位が相違していれば、S520に進む。尚、単位の相違のチェックは、信号オブジェクト($\alpha 20$)の‘単位へのポインタ’で示される「単位」内の‘単位名’と、受信端子に対応する「端子」の‘単位へのポインタ’で示される「単位」内の‘単位名’とを比較することで行われる(図3(e)参照)。また、以下の説明では、信号オブジェクトのデータ属性を、単位1、データ型1、LSB1、OFFSET1とし、また、データの内部値をN1とし、データの外部値をG1とする。そして、受信端子のデータ属性を、単位2、データ型2、LSB2、OFFSET2とする。

【0161】S520では、単位1と単位2とが異なっていることから、単位1から単位2への変換式を取り出す。尚、この変換式は予め用意されているものであり、信号オブジェクトの‘単位へのポインタ’で示される「単位」内の‘単位xへの変換式’(図3(e)参照)の中から、上記単位2へ変換するための式が抽出される。

【0162】そして、続くS530にて、データの外部値(以下、単に、外部値ともいう)G1を、S520で取り出した単位変換式に従って、単位2に応じた外部値

G2に変換する。例えば、単位変換式が「 $f(X) = A * X + B$ 」であるとすれば、単位2に応じた変換後の外部値G2は、下記の式1のようになる。

【0163】

【数1】 $G2 = A * G1 + B$ …式1

そして、このようにS530で外部値の変換を行うか、或いは、S510で信号オブジェクト側の単位1と受信端子側の単位2とが相違していない（一致している）と判定した場合には、S540に進んで、データの外部値（S530の処理が行われた場合には式1のG2であり、そうでなければ $G1 = G2$ である）を、受信端子側のLSB2及びOFFSET2で表した値を計算する。そして更に、続くS550にて、S540での計算結果が、受信端子側のデータ型2で表現できるか否かを判定し、データ型2で表現できなければ、続くS560にて、表示画面Hに、オーバーフローの警告を表示させる。

【0164】また、このS560の処理を行うか、或いは、S550にて、S540での計算結果が受信端子側のデータ型2で表現できる（オーバーフローしない）と判定した場合には、S570に進む。そして、このS570にて、単位変換式とLSB・OFFSET変換式とを組み合わせた変換式（以下、組合せ変換式という）を作成し、その組合せ変換式をデータの内部値（以下、単

$$N2 = (A * LSB1 * N1 + A * OFFSET1 + B - OFFSET2) / LSB2 \quad \text{…式5}$$

そして、S570にて、上記式5により内部値N1を受信端子側のデータ属性に応じた内部値N2に変換した後、続くS580にて、信号オブジェクトの「データの外部値」と「データの内部値」とに、夫々、S530で変換した外部値G2と、S570で変換した内部値N2とを設定する。そして更に、信号オブジェクトの「単位へのポインタ」に単位2へのポインタを設定すると共に、「データ型」にデータ型2を、「LSB」にLSB2を、「OFFSET」にOFFSET2を、夫々設定して、その後、当該データ属性変換処理を終了する。

【0171】このようなデータ属性変換処理が実行されることにより、シミュレーション時において、データ信号の発信端子のデータ属性と、その信号の受信端子のデータ属性とが相違していても、換言すれば、流される信号（データ信号）のデータ属性が受信側で取り扱われる信号のデータ属性と違っていても、その信号のデータ属性、外部値、及び内部値が、受信側のデータ属性に適合するものに自動的に変換されるのである。

【0172】次に、このようなシミュレータの作用について、図2に示した回路に対しテストツールを図21の如く接続した場合を例に挙げて説明する。尚、ここでは、図2と図21を合わせて参照されたい。

（1）まず、S300の処理により、信号発生テストツールST1のT端子を発信端子とし、基本素子（ストア素子）KS1のE端子E3を受信端子とし、信号種別を

に、内部値ともいう）N1に適用することにより、その内部値N1を受信端子側のデータ属性に応じた内部値N2に変換する。

【0165】ここで、組合せ変換式について説明する。

尚、組合せ変換式の作成に用いる単位変換式としては、S510で肯定判定された場合には、S520で取り出されたものが用いられる。また、S510で否定判定された場合には、「 $f(X) = X$ 」となるため、後述する式5にて $A = 1$ 、 $B = 0$ となる。

【0166】まず、LSBとOFFSETの定義（意味）により、次の式2、式3が成立する。

【0167】

【数2】 $G1 = LSB1 * N1 + OFFSET1$ …式2

【0168】

【数3】 $N2 = (G2 - OFFSET2) / LSB2$ …式3

そして、単位変換式が前述の例と同様に「 $f(X) = A * X + B$ 」であるとすれば、式1と式2より下記の式4が得られ、この式4を式3に代入することにより、組合せ変換式としての式5が得られる。

【0169】

【数4】 $G2 = A * LSB1 * N1 + A * OFFSET1 + B$ …式4

【0170】

【数5】

トリガ信号とした信号オブジェクトが生成されて、信号スタックに登録される。

【0173】（2）すると、S320、S360の処理により、基本素子KS1に定義された「シミュレーション動作」のうち、そのE端子E3に対応する動作（即ち、E端子E3にトリガ信号が来ると、A端子A5からアクセス信号を出力するという動作）が起動・実行され、基本素子KS1のA端子A5を発信端子とし、組合せ素子KS3のD端子D3を受信端子とし、信号種別をアクセス信号とした信号オブジェクトが生成される。

【0174】（3）すると今度は、S320、S370の処理により、上記D端子D3を発信端子とし、組合せ素子KS3を構成する基本素子（乗算素子）KS4のD端子D4を受信端子とし、信号種別をアクセス信号とした信号オブジェクトが生成される。

【0175】（4）そして次に、S320、S360の処理により、基本素子KS4に定義された「シミュレーション動作」のうち、そのD端子D4に対応する動作（即ち、D端子D4にアクセス信号が来ると、A端子A10とA端子A11から夫々アクセス信号を出力するという動作）が起動・実行される。

【0176】これにより、上記A端子A10を発信端子とし、組合せ素子KS3のA端子A7を受信端子とし、信号種別をアクセス信号とした信号オブジェクトと、上記A端子A11を発信端子とし、組合せ素子KS3を構

成する基本素子（加算素子）KS5のD端子D5を受信端子とし、信号種別をアクセス信号とした信号オブジェクトとが、夫々生成される。

【0177】（5）すると、例えば、まず、上記（4）で生成された2つの信号オブジェクトのうち、上記A端子A7を受信端子とする方の信号オブジェクトに対して、S320、S370の処理が実行され、上記A端子A7を発信端子とし、信号発生テストツールST2のD端子を受信端子とし、信号種別をアクセス信号とした信号オブジェクトが生成され、次に、S320、S360の処理が実行されることにより、信号発生テストツールST2に定義された‘シミュレーション動作’（即ち、D端子にアクセス信号が来ると所定値のデータ信号を返すという動作）が起動・実行される。

【0178】これにより、信号発生テストツールST2のD端子を発信端子とし、組合せ素子KS3のA端子A7を受信端子とし、信号種別をデータ信号とし、そのデータ値（外部値及び内部値）を所定値とした信号オブジェクトが生成され、その後、S320、S370の処理により、この信号オブジェクトと同じデータ値を持つ信号オブジェクトが、上記A端子A7を発信端子とし、基本素子KS4のA端子A10を受信端子として生成される。

【0179】（6）一方、上記（4）で生成された2つの信号オブジェクトのうち、基本素子KS5のD端子D5を受信端子とする方の信号オブジェクトについても、上記（4）、（5）のような処理が繰り返される。そして、これにより、基本素子KS5及び信号発生テストツールST3、ST4に夫々定義された‘シミュレーション動作’が起動・実行され、やがて、基本素子KS5のD端子D5を発信端子とし、基本素子KS4のA端子A11を受信端子とし、信号種別をデータ信号とし、そのデータ値を基本素子KS5の演算結果とした信号オブジェクトが生成される。

【0180】（7）すると、基本素子KS4に定義された‘シミュレーション動作’のうち、そのA端子A10、A11に対応するシミュレーション動作（即ち、A端子A10とA端子A11とに両方とも応答する2つのデータ信号が来ると、その2つのデータ値の積をD端子D4から返すという動作）が起動・実行される。

【0181】これにより、基本素子KS4のD端子D4を発信端子とし、組合せ素子KS3のD端子D3を受信端子とし、信号種別をデータ信号とし、そのデータ値を基本素子KS4の演算結果とした信号オブジェクトが生成され、更にその後、この信号オブジェクトと同じデータ値を持つ信号オブジェクトが、上記D端子D3を発信端子とし、モニタテストツールMT1の端子を受信端子として生成される。

【0182】（8）すると、モニタテストツールMT1の‘シミュレーション動作’に定義された動作が起動・

実行されて、基本素子KS4での演算結果が表示装置Hに表示される。

（9）一方、このようなシミュレーション時において、生成される信号オブジェクトの信号種別がデータ信号の場合には、図25に示したデータ属性変換処理が実行される（S330：YES）。そして、この処理により、発信端子と受信端子のデータ属性が異なる場合には、信号オブジェクトのデータ属性とそのデータ値（外部値及び内部値）とが、受信端子側のデータ属性に適合するものに自動変換される。

【0183】以上のように、シミュレータでは、回路モデルを構成する各基本素子を抽出して、その各基本素子に夫々設定された‘シミュレーション動作’を特定すると共に、その各‘シミュレーション動作’を、各基本素子同士の端子の機能に基づく結線状態に従い、信号発生テストツールによりトリガ信号（或いはアクセス信号）が発せられる端子を有した基本素子の‘シミュレーション動作’から順次実行するようにしている。そして、モニタテストツールが接続された基本素子の‘シミュレーション動作’の実行結果を、表示画面Hに表示させるようにしている。

【0184】このシミュレータによれば、表示画面H（詳しくは、回路編集画面21）に表示された回路図中の何れかのE端子又はD端子に、トリガ信号又はアクセス信号を発生する信号発生ツールを接続し、また、上記表示された回路図中の回路素子のうち、その処理内容の実行結果をモニタしたい回路素子の端子に、モニタテストツールを接続して、図24に示した処理を開始させれば、モニタテストツールの接続された基本素子が表す処理内容の実行結果が、表示画面Hに表示される。よって、作業者は、その表示を見ることにより、所望の処理が正しく行われるか否かを検証することができる。

【0185】従って、このシミュレータによれば、回路図の全体によって表される処理内容のうち、任意の部分の処理内容が思うように正しく行われるか否かを、後述するプログラムジェネレータによりプログラムを自動生成する前に、事前にチェックすることができ、所望のプログラムを作成する際のトータルの作業効率を向上させることができる。

【0186】しかも、このシミュレータによれば、前述した図25のデータ属性変換処理により、互いに接続される2つの回路素子にて、データ信号の発信端子と受信端子との、両者のデータ属性が異なる場合には、伝達されるデータ信号のデータ属性を、受信端子のデータ属性に自動変換するようにしている。

【0187】従って、当該装置の使用者は、各回路素子間でやり取りするデータの単位、データ型、LSB、及びOFFSETの適合性を厳しくチェックする必要がなく、作業効率をより一層向上させることができる。しかも、このシミュレータでは、D端子から出力されるデー

タ信号の外部値をA端子のデータ属性に適合するLSB及びOFFSETで表した値を計算し、その値がA端子側のデータ型で表現しきれない場合（オーバーフローする場合）には、表示画面Hに警告を表示させるようにしている（S540、S550）。よって、使用者は、D端子とA端子との間でデータ属性の不適合が発生していることを見逃すこともない。次に、回路エディタにより表示画面H上で描画された回路図が表す処理内容を、コンピュータに実行させるためのC言語によるプログラム（ANSI-Cのプログラム）を生成するプログラムジェネレータについて説明する。

【0188】まず、前述したように、使用者がキーボード5又はマウス7を操作してプログラムジェネレータソフトを起動すると、当該支援装置1はプログラムジェネレータとして動作を開始する。そして、キーボード5により、プログラム生成の対象としたい回路図に対応する回路モデルのファイル名（定義素子ファイル名）を入力し、所定の入力操作を行えば、その回路モデルが読み込まれて、その回路モデルに対し後述する図27～図29の処理が実行されることにより、プログラムが自動生成される。

【0189】ここで、プログラムジェネレータで行われる処理の概要について説明する。前述したように、本支援装置1で扱う回路図（データ回路）及び回路図データ（回路モデル）には、処理の制御を伝えるトリガ信号／完了信号と、データの授受を伝えるアクセス信号／データ信号の2組の信号がある。

【0190】そして、トリガ信号／完了信号に対する回路素子の基本的な動作は、次のようである。

- ・トリガ信号／完了信号を受信した組合せ素子のE端子／T端子は、受信した信号を逆側の接続先端子に伝達する。

【0191】・トリガ信号／完了信号を受信した基本素子のE端子／T端子は、その信号に対応する処理を行う。そして、処理は‘生成プログラム’として基本素子毎に定義されている（図3（f）参照）。

また、トリガ信号／完了信号の特徴は、次のようである。

【0192】・個々のトリガ信号に対しては、それに対応する完了信号が必ず返される。

- ・トリガ信号／完了信号は、制御を伝えるだけであり、データを伝えることはない。

このため、本支援装置1では、トリガ信号／完了信号と、生成するプログラム構造との関係を、次のようにしている。即ち、トリガ信号の伝達を、伝達先の端子に対応する関数を呼び出すことととらえ、完了信号の伝達を、その関数から戻ることととらえている。そして、トリガ信号／完了信号は、データを伝えないので、この関数には引数や戻り値がない。

【0193】一方、アクセス信号／データ信号に対する

回路素子の基本的な動作は、次のようである。

- ・アクセス信号／データ信号を受信した組合せ素子のD端子／A端子は、受信した信号を逆側の接続先端子に伝達する。

【0194】・アクセス信号／データ信号を受信した基本素子のD端子／A端子は、その信号に対応する処理を行う。そして、処理は‘生成プログラム’として基本素子毎に定義されている。

また、アクセス信号／データ信号の特徴は、次のようである。

【0195】・個々のアクセス信号に対しては、それに対応するデータ信号が必ず返される。

- ・データ信号は、データを伝えるが、アクセス信号は、データの要求を伝えるだけであり、データそのものを伝えることはない。

【0196】このため、本支援装置1では、アクセス信号／データ信号と、生成するプログラム構造との関係を、次のようにしている。即ち、アクセス信号の伝達を、伝達先の端子に対応する関数を呼び出すことととらえ、データ信号の伝達を、その関数から戻ることととらえている。そして、アクセス信号はデータを伝えないが、データ信号はデータを伝えるので、この関数には引数はないが戻り値があり、この戻り値がデータ信号の伝えるデータに対応する。

【0197】以上のことより、本プログラムジェネレータの生成するプログラムは、各端子に対応する関数の集まりとなる。そして、組合せ素子の端子に対応する関数は、トリガ信号／アクセス信号の伝達先の端子に対応する関数を呼び出す関数であり、基本素子の端子に対応する関数は、その基本素子に定義されている処理を行う関数である。尚、以下の説明では、このように各端子に対応する関数を「回路動作関数」と呼ぶことにする。

【0198】そして、各基本素子の動作（処理内容）を表わす‘生成プログラム’は、その基本素子の各端子に対応する関数の形で用意されており、E端子／D端子に対応する関数には、その端子に信号が来たときの動作が既述されている。そして、この動作の中で、T端子／A端子から信号を送信する場合は、その端子に対応する関数を呼び出すこととなる。また、T端子／A端子に対応する関数は、その接続先の端子に対する関数を呼び出すものである。

【0199】一方、本プログラムジェネレータでは、生成したプログラムの性能を良くするために、信号を中継するだけの端子などの一部の端子の回路動作関数は、マクロの形で生成する。そして、図26に示すように、マクロはヘッダファイルに出力し、関数はプログラムファイル（ソースファイル）に出力する。つまり、信号を中継する端子を削除するようにして、プログラムの性能向上を図っている。尚、図26は、当該支援装置1がプログラムジェネレータとして作動する場合の各部の関係を

概念的に表すブロック図である。

【0200】また、本プログラムジェネレータでは、接続されている端子間のデータ属性が異なる場合には、データ属性の変換を行うためのプログラムコードを自動生成するようにしている。以上のような基本方針に基づき、本プログラムジェネレータでは、プログラム生成対象の回路図を構成する回路素子の各端子について、回路モデルの「端子」や「信号線」の情報に基づき、その端子の種別（E端子、T端子、D端子、A端子）と接続先端子を判別すると共に、その端子に対応する関数を回路素子データベース内の各「基本素子」の「生成プログラム」から読み出し、その読み出した関数を結合してなるC言語のプログラムを構築するようにしている。

【0201】そこで次に、本プログラムジェネレータ（プログラムジェネレータソフト）で行われるプログラム生成のための処理について、図27～図29を用いて具体的に説明する。尚、図27は、プログラムジェネレータの処理の前半部を表すフローチャートであり、図28は、その後半部を表すフローチャートである。また、図29は、図28の処理中で実行されるデータ属性変換処理を表すフローチャートである。また、以下の説明では、回路素子を単に回路とも呼び、プログラム生成の対象となる回路図を階層的に構成する回路素子のうちで最も上位の回路素子、即ち最も外側の回路枠を有する回路素子を、生成開始回路と呼ぶことにする。

【0202】図27及び図28に示すように、プログラムジェネレータでは、生成開始回路を最上位の階層として下位の階層の順に、その各階層を順次、処理対象階層とすると共に、その処理対象階層の全ての各回路について、S640～S830の処理を行うようにしている。

【0203】即ち、まずS600にて、生成開始回路を処理対象階層として設定する。そして、続くS610にて、全ての階層について処理が終了したか否かを判定し、全階層について処理が終了すれば、当該プログラムジェネレータの処理を終えるが（S610：YES）、そうでなければ、S620に進んで、現在、処理対象階層として設定されている階層の全ての回路について、S640～S830の処理が終了したか否かを判定する。

【0204】そして、S620で肯定判定した場合には、S630に進んで、処理対象階層を1つ下位の階層に設定し直して、S610に戻る。一方、S620で否定判定した場合には、処理対象階層の回路のうちの1つを処理対象回路として設定し、その処理対象回路に対してS640以降の処理を行う。尚、生成開始回路が処理対象階層である場合には、当然ながら生成開始回路が処理対象回路として設定される。

【0205】以下、S640以降の処理について説明する。まず、S640～S680の処理により、処理対象回路の全てのE端子について（S640）、以下の処理を繰り返す。即ち、処理対象回路の種類を判定し（S6

50）、その判定結果に応じて、S660～S680の何れかの処理を行う。

【0206】まず第1に、処理対象回路が生成開始回路の場合には（S660）、当該E端子の接続先の端子（接続先端子）の回路動作関数を呼び（コールする）関数を出力する。接続先の端子がない場合は、エラーとなる。尚、このように出力される関数は、前述したようにプログラムファイルに出力されて記憶され、以下の説明でも同様である。

【0207】また第2に、処理対象回路が組合せ素子の場合には（S670）、当該E端子の回路動作関数を接続先の端子の回路動作関数に置き換えるマクロを出力する。接続先の端子がない場合には、エラーとなる。尚、このように出力されるマクロは、前述したようにヘッダファイルに出力されて記憶され、以下の説明でも同様である。

【0208】また第3に、処理対象回路が基本素子の場合には（S680）、その基本素子に設けられた「生成プログラム」の中で、当該E端子の動作に対応する部分の関数を出力する。次に、全てのE端子について上記の処理が終わると（S640：YES）、S690～S720の処理により、処理対象回路の全てのT端子について（S690）、以下の処理を繰り返す。

【0209】即ち、処理対象回路の種類を判定し（S700）、その判定結果に応じて、S710或いはS720の処理を行う。まず第1に、処理対象回路が生成開始回路の場合には（S710）、リターンするだけの関数を出力する。

【0210】また第2に、処理対象回路が生成開始回路以外の場合には（S720）、当該T端子の回路動作関数を接続先の端子の回路動作関数に置き換えるマクロを出力する。接続先の端子がない場合は、当該T端子の回路動作関数を空文に置き換えるマクロを出力する。

【0211】次に、全てのT端子について上記の処理が終わると（S690：YES）、S730～S760の処理により、処理対象回路の全てのD端子について（S730）、以下の処理を繰り返す。即ち、処理対象回路の種類を判定し（S740）、その判定結果に応じて、S750、S770、及びS780のうちの何れかの処理を行う。

【0212】まず第1に、処理対象回路が生成開始回路の場合には（S750）、当該D端子の接続先の端子の回路動作関数を呼び、その戻り値を返す関数を出力する。接続先がない場合には、エラーとなる。そして更に、この場合には、続くS760で後述するデータ属性変換処理（図29）を実行し、当該D端子のデータ属性と接続先の端子のデータ属性が異なる場合は、データ属性変換のための関数（以下、データ属性変換関数という）を作成して出力すると共に、接続先の端子の回路動作関数を、その回路動作関数に上記データ属性変換関数

をかぶせたものに置き換えるマクロを出力する。

【0213】また第2に、処理対象回路が組合せ素子の場合には(S770)、当該D端子の回路動作関数を接続先の端子の回路動作関数に置き換えるマクロを出力する。接続先の端子がない場合は、エラーとなる。そして、この場合にも、S760に進んで、後述するデータ属性変換処理(図29)を実行する。

【0214】また第3に、処理対象回路が基本素子の場合には(S780)、その基本素子に設けられた'生成プログラム'の中で、当該D端子の動作に対応する部分の関数を出力する。次に、全てのD端子について上記の処理が終わると(S730:YES)、S790~S830の処理により、処理対象回路の全てのA端子について(S790)、以下の処理を繰り返す。

【0215】即ち、処理対象回路の種類を判定し(S800)、その判定結果に応じて、S810或いはS820の処理を行う。まず第1に、処理対象回路が生成開始回路の場合には(S810)、デフォルト値である0を返すだけの関数を出力する。

【0216】また第2に、処理対象回路が生成開始回路以外の場合には(S820)、当該A端子の回路動作関数を接続先の端子の回路動作関数に置き換えるマクロを出力する。接続先の端子がない場合は、当該A端子の回路動作関数をデフォルト値である0に置き換えるマクロを出力する。そして更に、この場合には、続くS830で後述するデータ属性変換処理(図29)を実行し、当該A端子のデータ属性と接続先の端子のデータ属性が異なる場合は、データ属性変換関数を作成して出力すると共に、接続先の端子の回路動作関数を、その回路動作関数に上記データ属性変換関数をかぶせたものに置き換えるマクロを出力する。

【0217】このような処理が全てのA端子について終わると(S790:YES)、S840に移行して、現在の処理対象階層における次の回路を処理対象回路として設定し直し、その後、S620に戻る。すると、その新たな処理対象回路について、上記S640~S830の処理が行われることとなる。

【0218】ここで、前述したS760及びS830で夫々実行されるデータ属性変換処理について、図29に沿って説明する。図29に示すように、データ属性変換処理の実行が開始されると、まず、S900にて、現在の処理対象である端子(即ち、D端子又はA端子であって、以下、処理対象端子という)のデータ属性と、接続先の端子(接続先端子)のデータ属性とが異なっている

$$g(N) = (A * LSB1 * N + A * OFFSET1 + B - OFFSET2) / LSB2 \quad \cdots \text{式6}$$

そして、続くS940にて、S930で作成した変換式を関数として出力する。このように出力される関数が前述したデータ属性変換関数であり、更に続くS950にて、接続先端子の回路動作関数を、その回路動作関数に上記データ属性変換関数をかぶせた(合成した)ものに

可否かを判定し、両端子のデータ属性が一致していれば(S900:NO)、そのまま当該処理を終了する。

【0219】これに対して、上記両端子のデータ属性が異なっている場合には(S900:YES)、続くS910にて、上記両端子の単位が相違しているか否かを判定する。そして、単位が相違していれば、S920に進む。尚、単位の相違のチェックは、処理対象端子に対応する「単位」の「単位へのポインタ」で示される「単位」内の「単位名」と、接続先端子に対応する「端子」の「単位へのポインタ」で示される「単位」内の「単位名」とを比較することで行われる(図3(e)参照)。また、以下の説明では、シミュレータで実行されるデータ属性変換処理(図25)に関する説明の場合と同様に、処理対象端子のデータ属性を、単位1、データ型1、LSB1、OFFSET1とし、接続先端子のデータ属性を、単位2、データ型2、LSB2、OFFSET2とする。

【0220】S920では、単位1と単位2とが異なっていることから、単位1から単位2への変換式を取り出す。尚、この変換式は予め用意されているものであり、処理対象端子に対応する「単位」の「単位へのポインタ」で示される「単位」内の「単位xへの変換式」(図3(e)参照)の中から、上記単位2へ変換するための式が抽出される。

【0221】そして、このようにS920で単位変換式を取り出すか、或いは、S910で処理対象端子の単位1と接続先端子の単位2とが相違していない(一致している)と判定した場合には、S930に進んで、処理されるデータの内部値を接続先端子のデータ属性に適合する値に変換するための変換式を作成する。

【0222】即ち、この変換式は、単位変換式とLSB・OFFSET変換式とを組み合わせた組合せ変換式である。そして、単位変換式が「 $f(X) = A * X + B$ 」であり、処理対象端子側のデータの内部値がNであるとすると、この組合せ変換式g(N)は、前述した式5より下記の式6となる。尚、単位変換式「 $f(X) = A * X + B$ 」としては、S910で肯定判定された場合には、S920で取り出されたものが用いられる。また、S910で否定判定された場合には、「 $f(X) = X$ 」となるため、下記の式6において $A = 1$ 、 $B = 0$ となる。

【0223】

【数6】

置き換えるマクロ(以下、データ属性変換適用マクロという)を出力する。そして、このS950の処理を行った後、当該データ属性変換処理を終了する。

【0224】例えば、処理対象端子のデータ属性(即ち、変換元のデータ属性)を、単位=km、データ型=

short (=2バイト), LSB=8, OFFSET=0とし、また、接続先端子のデータ属性(即ち、変換先のデータ属性)を、単位=m, データ型=int (=4バイト), LSB=128, OFFSET=1000とすると、単位変換式は「 $f(X) = 1000 * X$ 」となるため、S940で出力されるデータ属性変換関数は、上記式6から下記「」内の式7のようになる。

【0225】

【数7】

```
「int changel (short X) {
    return (int) ((8000*X-1000) /128) ;
}」 …式7
```

そして、データ属性変換適用マクロは、下記「」内の式8のようになる。

【0226】

【数8】

```
「#define D1 () changel (D1 ())」 …式8
```

尚、上記式7及び式8は、周知のC言語による表現であるが、その内容を簡単に説明すると、まず、上記式7は、changel () というデータ属性変換関数が、2バイトのXというデータを、 $((8000 * X - 1000) / 128)$ という演算を行って4バイトで返す、という関数であることを表している。そして、式8は、D1 () という関数を、その関数に上記関数 changel () をかぶせたものに置き換える、という内容を表しており、上記関数 D1 () が、接続先端子の元の回路動作関数に相当している。

【0227】次に、以上のようなプログラムジェネレータの作用について、図2(a)に示した回路図をプログラムの生成対象とした場合を例に挙げ、図30～図33を用いて説明する。ここで、図30は、この例において、ヘッダファイルに出力されたマクロを表しており、図31は、同じくこの例において、プログラムファイル(以下、ソースファイルという)に出力された関数を表している。そして、図32は、図31に示したソースファイルの前半部の内容を表わすフローチャートであり、図33は、図31に示したソースファイルの後半部の内容を表わすフローチャートである。また、図30において、符号 Hf1～Hf12は、ヘッダファイルに出力されたマクロの夫々を示すものであり、図31において、符号 Sf1～Sf14は、ソースファイルに出力された関数の夫々を示すものである。そして、図30及び図31において、例えば E5 (), T2 (), D1 (), A3 () といった具合に、E, T, D, Aの各アルファベットから始まるものは、その () の前の文字列が表わす端子(図2の何れかの端子)に対応した関数(回路動作関数)である。

【0228】この例の場合、まず、図2のT端子T3, T2については、図30のHf1, Hf3に示す如く、図27におけるS720の処理により、その端子の回路

動作関数を接続先端子の回路動作関数に置き換えるマクロが出力される。例えば、T端子T3については、図30のHf1に示されているように、その回路動作関数 T3 () を接続先であるE端子E5の回路動作関数 E5 () に置き換えるマクロが出力される。

【0229】また、図2のD端子D3については、図30のHf5に示す如く、図28におけるS770の処理により、その端子の回路動作関数 D3 () を接続先であるD端子D4の回路動作関数 D4 () に置き換えるマクロが出力される。また、図2のA端子A5～A13については、図30のHf2, Hf4, Hf6～Hf12に示す如く、図28におけるS820の処理により、その端子の回路動作関数を接続先端子の回路動作関数に置き換えるマクロが出力される。例えば、A端子A5については、図30のHf2に示されているように、その回路動作関数 A5 () を接続先であるD端子D3の回路動作関数 D3 () に置き換えるマクロが出力される。

【0230】一方、図2のE端子E1については、図31のSf1及び図32(a)に示す如く、図27におけるS660の処理により、接続先であるE端子E3の回路動作関数 E3 () を呼ぶ関数 E1 () が出力される。同様に、図2のE端子E2については、図31のSf2及び図32(b)に示す如く、図27におけるS660の処理により、接続先であるE端子E4の回路動作関数 E4 () を呼ぶ関数 E2 () が出力される。

【0231】また、図2のT端子T1については、図31のSf3及び図32(c)に示す如く、図27におけるS710の処理により、リターンするだけの関数 T1 () が出力される。また更に、図2のD端子D1については、図31のSf4及び図32(d)に示す如く、図28におけるS750の処理により、接続先であるD端子D2の回路動作関数 D2 () を呼び、その戻り値(結果のデータ)を返す関数 D1 () が出力される。

【0232】そして、図2のA端子A1～A4については、図31のSf5～Sf8及び図32(e)～(h)に示す如く、図28におけるS810の処理により、0を返すだけの関数 A1 ()～A4 () が出力される。尚、ここで0を返すようにしているのは、図31のようなソースファイルが出力された後で、作業者が、0に代えて、任意の値或いは関数をあてはめれば良いからである。

【0233】一方、図2のE端子E3については、この端子を有する回路素子が基本素子KS1であるため、図27におけるS680の処理により、この基本素子KS1に設けられた‘生成プログラム’の中で当該E端子E3に対応する部分の関数が出力される。即ち、図31のSf9に示す如く、A端子A5の回路動作関数 A5 () を呼んで、その戻り値をDataという領域にストアし、次にT端子T3の回路動作関数 T3 () を呼ぶ、という関数 E3 () が出力される。

【0234】そして、この場合、図30のHf2、Hf5、Hf1に夫々示されるマクロにより、A5()はD3()に置き換えられ、D3()はD4()に置き換えられ、T3()はE5()に置き換えられるため、結局、当該E端子E3に対応する関数E3()の内容は、図32(i)に示すように、当該関数E3()が呼ばれると、D端子D4の回路動作関数D4()を呼んで、その戻り値をストアし、次にE端子E5の回路動作関数E5()を呼び、関数E5()の実行が完了すると、当該関数E3()からリターンする、といった処理となる。

【0235】尚、この処理は、基本素子KS1の動作のうち「E端子E3にトリガ信号が来ると、A端子A5からアクセス信号を出力し、それに応答するデータ信号が来ると、そのデータ値を内部にストアし、次にT端子T3からトリガ信号を出力し、それに対応する完了信号が来ると、E端子E3から完了信号を返す」というE端子E3の動作に対応している。

【0236】次に、図2のE端子E4についても、この端子を有する回路素子が基本素子KS1であるため、図27におけるS680の処理により、この基本素子KS1に設けられた'生成プログラム'の中で当該E端子E4に対応する部分の関数が出力される。即ち、図31のSf10及び図33(a)に示す如く、Dataという領域に0をストアして戻る、という関数E4()が出力される。

【0237】尚、この関数E4()による処理は、基本素子KS1の動作のうち「E端子E4にトリガ信号が来ると、内部のストア値を0にして、E端子E4から完了信号を返す」というE端子E4の動作に対応している。また、図2のD端子D2については、この端子を有する回路素子が基本素子KS1であるため、図28におけるS780の処理により、この基本素子KS1に設けられた'生成プログラム'の中で当該D端子D2に対応する部分の関数が出力される。即ち、図31のSf11及び図33(b)に示す如く、Dataという領域にストアしている値を返して戻る、という関数D2()が出力される。

【0238】尚、この関数D2()による処理は、基本素子KS1の動作のうち「D端子D2にアクセス信号が来ると、内部のストア値をD端子D2から返す」というD端子D2の動作に対応している。一方、図2のE端子E5については、この端子を有する回路素子が基本素子KS2であるため、図27におけるS680の処理により、この基本素子KS2に設けられた'生成プログラム'の中で当該E端子E5に対応する部分の関数が出力される。即ち、図31のSf12に示す如く、A端子A6の回路動作関数A6()を呼び、その戻り値が0でなければT端子T2の回路動作関数T2()を呼ぶ、という関数E5()が出力される。

【0239】そして、この場合、図30のHf4、Hf

3に夫々示されるマクロにより、A6()はA4()に置き換えられ、T2()はT1()に置き換えられるため、結局、当該E端子E5に対応する関数E5()の内容は、図33(c)に示すように、当該関数E5()が呼ばれると、A端子A4の回路動作関数A4()を呼んで、その戻り値が0ならば、そのまま当該関数E5()からリターンし、関数A4()の戻り値が0以外ならば、T端子T1の回路動作関数T1()を呼び、関数T1()の実行が完了すると当該関数E5()からリターンする、といった処理となる。

【0240】尚、この処理は、基本素子KS2の動作のうち「E端子E5にトリガ信号が来ると、A端子A6からアクセス信号を出力し、それに応答するデータ信号が来ると、そのデータ値が0ならば、そのままE端子E5から完了信号を返し、逆に、上記データ値が0以外ならば、T端子T2からトリガ信号を出して、それに対応する完了信号が来ると、E端子E5から完了信号を返す」というE端子E5の動作に対応している。

【0241】また、図2のD端子D4については、この端子を有する回路素子が基本素子KS4であるため、図28におけるS780の処理により、この基本素子KS4に設けられた'生成プログラム'の中で当該D端子D4に対応する部分の関数が出力される。即ち、図31のSf13に示す如く、A端子A10の回路動作関数A10()とA端子A11の回路動作関数A11()とを呼び、その両戻り値を乗算して返す、という関数D4()が出力される。

【0242】そして、この場合、図30のHf9、Hf6、Hf10に夫々示されるマクロにより、A10()はA7()に置き換えられ、A7()はA1()に置き換えられ、A11()はD5()に置き換えられるため、結局、当該D端子D4に対応する関数D4()の内容は、図33(d)に示すように、当該関数D4()が呼ばれると、A端子A1の回路動作関数A1()とD端子D5の回路動作関数D5()とを呼び、その両戻り値を乗算して返した後リターンする、といった処理となる。

【0243】尚、この処理は、基本素子KS4の動作のうち「D端子D4にアクセス信号が来ると、A端子A10とA端子A11から夫々アクセス信号を出力し、それらに夫々応答する2つのデータ信号が来ると、その2つのデータ値の積をD端子D4から返す」というD端子D4の動作に対応している。

【0244】また同様に、図2のD端子D5についても、この端子を有する回路素子が基本素子KS5であるため、図28におけるS780の処理により、この基本素子KS5に設けられた'生成プログラム'の中で当該D端子D5に対応する部分の関数が出力される。即ち、図31のSf14に示す如く、A端子A12の回路動作関数A12()とA端子A13の回路動作関数A1

3 () とを呼び、その両戻り値を加算して返す、という関数 D5 () が出力される。

【0245】そして、この場合、図30のHf11、Hf7、Hf12、Hf8に夫々示されるマクロにより、A12 () はA8 () に置き換えられ、A8 () はA2 () に置き換えられ、A13 () はA9 () に置き換えられ、A9 () はA3 () に置き換えられるため、結局、当該D端子D5に対応する関数D5 () の内容は、図33 (e) に示すように、当該関数D5 () が呼ばれると、A端子A2の回路動作関数A2 () とA端子A3の回路動作関数A3 () とを呼び、その両戻り値を加算して返した後リターンする、といった処理となる。

【0246】尚、この処理は、基本素子KS5の動作のうち「D端子D5にアクセス信号が来ると、A端子A12とA端子A13から夫々アクセス信号を出力し、それらに夫々応答する2つのデータ信号が来ると、その2つのデータ値の和をD端子D5から返す」というD端子D5の動作に対応している。

【0247】そして、以上説明した図30のヘッダファイル及び図31のソースファイルからなるC言語のプログラムは、図2 (a) に例示した回路の動作をコンピュータに実行させることができるものとなる。以上のようにプログラムジェネレータでは、回路モデルを構成する各基本素子を抽出し、回路素子データベース (図3 (f) の「基本素子」) を参照することで、その各基本素子の「生成プログラム」内に関数の形で用意されたプログラム (つまり、回路モデルに対応する回路図により表現されたプログラムの各機能部分として用いられる小プログラム) を特定すると共に、その各プログラムを、各基本素子同士の端子の機能に基づく結線状態に従って結合することにより、回路モデルに対応した回路図の処理内容を表わすC言語のプログラムを、構築・生成するようにしている。

【0248】このプログラムジェネレータによれば、回路エディタにより前述したように所望の処理内容を表わす回路図を描画し、その回路図に対応する回路モデルに対して当該プログラムジェネレータを起動させることにより、上記描画した回路図が表わす処理内容をコンピュータに実行させるためのプログラムを自動生成することができる。

【0249】このように、本実施例の支援装置1によれば、作成しようとするプログラムにおける各処理同士の関係や、各処理間でのデータ授受の関係を、2次元に見ることができ上記各関係を総合的に理解し易い回路図を描画することによって、所望のプログラムを自動的に作成することができる。よって、ソフトウェアに関する知識に精通していなくても所望のプログラムを効率良く作成することができるようになる。

【0250】しかも、本支援装置1がプログラムジェネレータとして機能する際には、前述した図29のデータ

属性変換処理が実行され、データの授受を行う端子同士のデータ属性 (単位、データ型、LSB、及びOFFSET) が異なる場合には、データをその受信側のデータ属性に適合させるためのデータ属性変換用プログラムコード (データ属性変換適用マクロ) が自動生成される。

【0251】従って、プログラムを作る際に、各回路素子間でやり取りするデータの単位、データ型、LSB、及びOFFSETの適合性を厳しくチェックする必要がなく、作業効率をより一層向上させることができる。また、本支援装置1によれば、回路エディタにより回路素子間の結線を変えるだけで、各処理の実行順序や、制御データの授受関係を変更することができる。

【0252】しかも、本支援装置1によれば、プログラムジェネレータによりプログラムを生成する前に、プログラムの生成対象である回路に対してシミュレーションを行うことができるため、ソフト仕様の評価及び確認を事前に行うことができる。また更に、本支援装置1によれば、回路図をデータベース化することにより、作成したソフト仕様の資源化及び再利用化を行うことができる。

【0253】ここで、本実施例の支援装置1により、車両用内燃機関の燃料噴射時間 (有効噴射時間) を演算する演算プログラムを作成した場合の例について説明する。内燃機関の燃料噴射時間は、同期噴射時間と非同期噴射時間との和からなり、同期噴射時間は、内燃機関が始動時の場合に設定されるべき始動時噴射時間と、内燃機関が始動後の場合に設定されるべき始動後噴射時間との、何れかに設定されることが知られている。そこで、以下では、上記同期噴射時間を求める演算プログラムを生成するために描いたデータ回路について説明する。

【0254】まず、図34は、同期噴射時間TSYの演算プログラムをデータ回路で表した全体図である。尚、図34に例示する回路は、内燃機関の制御に関する他の演算プログラム (例えば、点火時期演算プログラムやアイドルスピードコントロールバルブの制御開度演算プログラムなど) を表す他の回路と共に、内燃機関の制御全体を表す上位回路の構成要素 (要素素子) となっている。

【0255】図34において、まず、回路素子KS10は、D端子D21にアクセス信号が来ると、内燃機関の始動回転数 (例えば、400rpm) NSTを表わすデータをD端子D21から返すものであり、回路素子KS11は、D端子D22にアクセス信号が来ると、3つのA端子A27、A28、A29から夫々アクセス信号を出力して、A端子A27から上記回路素子KS10のデータを取得し、A端子A28からエンジン回転数Nを表わすデータを取得し、A端子A29からスタータスイッチがONしているか否かを表わすスタータ信号STAを取得し、更に、その取得した各データに基づき、エンジン回転数Nが始動回転数NST以下で且つスタータスイッチがONであれば始動時であることを表わす始動時信号

を、そうでなければ非始動時信号をD端子D22から返すものである。

【0256】また、回路素子KS12は、E端子E22に所定時間毎の時間周期信号ETが来ると、A端子A30からアクセス信号を出力して、このA端子A30から冷却水温THWを表わすデータを取得すると共に、その冷却水温THWに基づき始動時噴射時間TSTを求め、また、D端子D23にアクセス信号が来ると、その求めた始動時噴射時間TSTを表わすデータをD端子D23から返すものである。

【0257】一方、回路素子KS13は、D端子D24にアクセス信号が来ると、始動後噴射時間TAFを表わすデータをD端子D24から返すものであり、図35に示すように複数の回路素子から構成されている。即ち、図35において、回路素子KS14は、D端子D27にアクセス信号が来ると、所定の定数K1を表わすデータをD端子D27から返すものであり、回路素子KS15は、D端子D28にアクセス信号が来ると、4つのA端子A41、A42、A43、A44から夫々アクセス信号を出力して、A端子A41から空気質量Gを表わすデータを取得し、A端子A42から目標空燃比(A/F)Tを表わすデータを取得し、A端子A43からエンジン回転数Nを表わすデータを取得し、A端子A44から上記回路素子KS14のデータを取得し、更に、それら取得したデータを用いて、基本噴射時間TPを「 $TP = G / N / (K1 * (A/F)T)$ 」なる演算式に基づき算出して、その基本噴射時間TPを表わすデータをD端子D28から返すものである。

【0258】また、回路素子KS16は、E端子E26に回転角信号EAが来ると、A端子A45、A47から夫々アクセス信号を出力して、A端子A45からエンジン回転数Nを表わすデータを取得し、A端子A47からスロットル開度TAを表わすデータを取得し、その取得した両データを用いて加減速運転時の補正係数FADを求め、また、E端子E27に時間周期信号ETが来ると、A端子A46、A48から夫々アクセス信号を出力して、A端子A46から冷却水温THWを表わすデータを取得し、A端子A48から酸素濃度センサ(O₂センサ)の検出値OXLを表わすデータを取得し、その取得した両データと上記求めた加減速運転時の補正係数FADとから基本噴射時間TPに対する補正係数FCを求め、更に、D端子D29にアクセス信号が来ると、上記求めた補正係数FCを表わすデータをD端子D29から返すものである。

【0259】そして、回路素子KS17は、図2の基本素子KS4と同じ乗算素子であり、D端子D30にアクセス信号が来ると、A端子A49、A50から夫々アクセス信号を出力して、A端子A49から回路素子KS15での基本噴射時間TPを表わすデータを取得し、A端子A50から回路素子KS16での補正係数FCを表わ

すデータを取得し、更に、その取得したデータを用いて基本噴射時間TPと補正係数FCとを乗算した値($TP * FC$)を求め、その乗算値を表わすデータをD端子D30から返すものである。

【0260】また、回路素子KS18は、図2の基本素子KS1と同じストア素子であり、E端子E28に時間周期信号ETが来ると、A端子A51からアクセス信号を出力して、回路素子KS17から上記乗算値($TP * FC$)を表わすデータを取得し、また、D端子D31にアクセス信号が来ると、その取得したデータを始動後噴射時間TAFを表わすデータとしてD端子D31から返すものである。

【0261】そして、図34において、回路素子KS19は、D端子D25にアクセス信号が来ると、A端子A37からアクセス信号を出力して、回路素子KS11からの信号を取得し、その信号が始動時信号であれば、A端子A38からアクセス信号を出力して、回路素子KS12から始動時噴射時間TSTを表わすデータを取得し、その取得したデータをD端子D25から返す。また、回路素子KS11からの信号が非始動時信号であれば、A端子A39からアクセス信号を出力して、回路素子KS13から始動後噴射時間TAFを表わすデータを取得し、その取得したデータをD端子D25から返す。

【0262】そして更に、図34において、回路素子KS20は、図2の基本素子KS1と同じストア素子であり、E端子E25に時間周期信号ETが来ると、A端子A40からアクセス信号を出力して、回路素子KS19からデータを取得し、また、D端子D26にアクセス信号が来ると、その取得したデータを同期噴射時間TSYを表わすデータとしてD端子D26から返すものである。

【0263】以上のようなデータ回路では、図34のE端子E21にトリガ信号としての時間周期信号ETを与えると、エンジン回転数Nが始動回転数NST以下で且つスタータスイッチがONであれば、始動時と判断されて、回路素子KS12で求められた始動時噴射時間TSTが同期噴射時間TSYとして回路素子KS20にストアされ、そうでなければ、始動後と判断されて、回路素子KS13(回路素子KS14~KS18)で求められた始動後噴射時間TAFが同期噴射時間TSYとして回路素子KS20にストアされ、図34のD端子D20にアクセス信号を与えると、回路素子KS20にストアされた同期噴射時間TSYがD端子D20から出力される、といった内容を表わすこととなる。

【0264】そして、内燃機関の燃料噴射時間を演算するという複雑な処理手順であっても、図34及び図35のようなデータ回路を回路エディタによって描画し、その回路モデルに対してプログラムジェネレータを作動させることにより、燃料噴射時間を演算するための演算プログラムを簡単に作成することができるのである。尚、本実施例においては、回路素子のE端子が第1の端子に

相当し、T 端子が第 2 の端子に相当し、D 端子が第 3 の端子に相当し、A 端子が第 4 の端子に相当している。そして、CD-ROM 15 内の回路素子データベースがインストールされた後のハードディスクユニット 11 が、素子情報記憶手段に相当している。

【0265】また、図 3 に示した回路モデル（回路図データ）のうち、「定義素子」の「回路枠座標」、「要素素子」の「配置座標」、「端子」の「配置座標」、及び「信号線」の「形状」と、これらの情報を関連付ける各オブジェクト内のポイントとが、所定の表示装置に疑似回路図を表示するための情報に相当しており、また、図 3 に示した回路モデルのうち、「定義素子」の内部に設定された各ポイント、「要素素子」、「信号線」、及び「端子」などが、疑似回路図中の仮想回路素子、端子、及び該端子の接続を特定するための索引情報に相当している。

【0266】また更に、プログラムジェネレータにて、キーボード 5 によりプログラム生成の対象としたい回路図に対応する回路モデルのファイル名が入力された際に、その回路モデルを読み込むために行われる処理が、プログラム作成装置の読込手段に相当している。そして、図 27 及び図 28 の処理がプログラム構築手段に相当し、図 29 の S950 で出力されるデータ属性変換適用マクロが、データ形態変換用プログラムに相当し、回路エディタがプログラム作成装置の描画手段に相当している。

【0267】一方、回路エディタにて、表示装置 9 の登録回路一覧表示画面 19 に登録回路（既に登録されている回路素子）のシンボル等を表示させるために行われる処理が、一覧表示手段に相当し、図 6 の処理のうち、図 10（オブジェクト選択）及び図 13（要素素子追加）の処理が素子選択手段に相当し、図 10 及び図 17（結線）の処理が結線手段に相当し、図 8（定義素子セーブ）の処理が記録制御手段に相当している。

【0268】また、シミュレータにおいて、キーボード 5 によりシミュレーションの対象としたい回路モデルのファイル名を入力した際に、その回路モデルを読み込むために行われる処理が、シミュレーション装置の読込手段に相当し、その読み込んだ回路モデルを表わす回路図を回路編集画面 21 に表示させるために行われる処理が、疑似回路図表示手段に相当している。

【0269】そして、マウス 7 の操作に応じて、回路編集画面 21 に表示されている回路内の端子と、登録テストツール一覧表示画面に表示されている信号発生テストツールの端子とをプローブにより接続するために行われる処理が、端子選択手段に相当し、マウス 7 の操作に応じて、回路編集画面 21 に表示されている回路内の端子と、登録テストツール一覧表示画面に表示されているモニタテストツールの端子とをプローブにより接続するために行われる処理が、モニタ対象選択手段に相当してい

る。また更に、図 24 の処理がプログラム実行手段に相当し、モニタテストツール及び図 24 の処理が、実行結果表示手段に相当している。

【図面の簡単な説明】

【図 1】 実施例の制御プログラム開発支援装置の構成を表す構成図である。

【図 2】 実施例の制御プログラム開発支援装置で用いる回路図を説明する説明図である。

【図 3】 回路図データ（回路モデル）を説明する説明図である。

【図 4】 回路エディタでの表示画面を説明する説明図である。

【図 5】 回路エディタを概念的に表したブロック図である。

【図 6】 回路エディタの処理全体を表すフローチャートである。

【図 7】 図 6 の S140 で実行される処理（定義素子オープン）を表すフローチャートである。

【図 8】 図 6 の S150 で実行される処理（定義素子セーブ）を表すフローチャートである。

【図 9】 図 6 の S160 で実行される処理（新定義素子作成）を表すフローチャートである。

【図 10】 図 6 の S170 で実行される処理（オブジェクト選択）を表すフローチャートである。

【図 11】 図 6 の S180 で実行される処理（オブジェクト選択解除）を表すフローチャートである。

【図 12】 図 6 の S190 で実行される処理（回路枠座標変更）を表すフローチャートである。

【図 13】 図 6 の S200 で実行される処理（要素素子追加）を表すフローチャートである。

【図 14】 図 6 の S210 で実行される処理（要素素子削除）を表すフローチャートである。

【図 15】 図 6 の S220 で実行される処理（端子追加）を表すフローチャートである。

【図 16】 図 6 の S230 で実行される処理（端子削除）を表すフローチャートである。

【図 17】 図 6 の S240 で実行される処理（結線）を表すフローチャートである。

【図 18】 図 6 の S250 で実行される処理（信号線削除）を表すフローチャートである。

【図 19】 図 6 の S260 で実行される処理（データ属性設定）を表すフローチャートである。

【図 20】 テストツール（シミュレーション専用回路素子）及びプローブのデータ構成を説明する説明図である。

【図 21】 テストツールの接続を説明する説明図である。

【図 22】 シミュレータを概念的に表したブロック図である。

【図 23】 信号オブジェクトのデータ構成を説明する

説明図である。

【図 24】 シミュレータの処理を表すフローチャートである。

【図 25】 図 24 で実行されるデータ属性変換処理を表すフローチャートである。

【図 26】 プログラムジェネレータを概念的に表したブロック図である。

【図 27】 プログラムジェネレータの処理前半部を表すフローチャートである。

【図 28】 プログラムジェネレータの処理後半部を表すフローチャートである。

【図 29】 図 28 で実行されるデータ属性変換処理を表すフローチャートである。

【図 30】 プログラムジェネレータにより出力されるヘッダファイルの一例を説明する説明図である。

【図 31】 プログラムジェネレータにより出力されるプログラムファイル（ソースファイル）の一例を説明する説明図である。

【図 32】 図 31 の前半部の内容を表わすフローチャートである。

【図 33】 図 31 の後半部の内容を表わすフローチャートである。

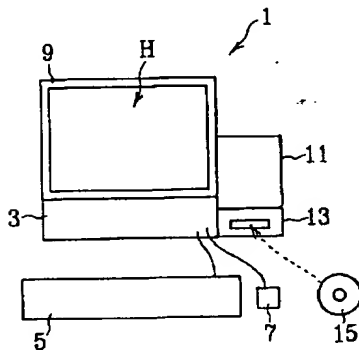
【図 34】 内燃機関の同期噴射時間の演算プログラムをデータ回路で表した図である。

【図 35】 始動後噴射時間の演算プログラムをデータ回路を表わした図である。

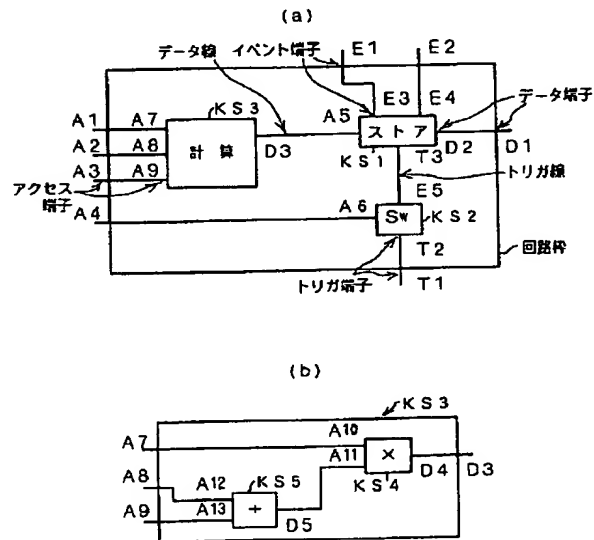
【符号の説明】

1…制御プログラム開発支援装置 3…コンピュータ本体
5…キーボード 7…マウス 9…表示装置
11…ハードディスクユニット 13…CD-ROM ドライブ
15…CD-ROM 17…メニュー画面 19…登録回路一覧表示画面
21…回路編集画面 KS1~KS5, KS10~KS20…回路素子

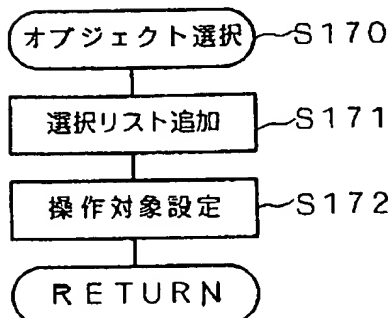
【図 1】



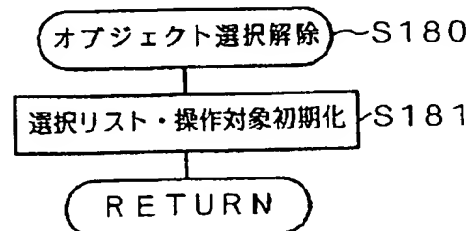
【図 2】



【図 10】



【図 11】



【図3】

(a) : 定義素子

定義素子名
回路枠座標
定義端子の個数 (n)
端子1へのポインタ
...
端子nへのポインタ
要素素子の個数 (m)
要素素子1へのポインタ
...
要素素子mへのポインタ
信号線の個数 (k)
信号線1へのポインタ
...
信号線kへのポインタ

(b) : 要素素子

要素素子名
配置座標
定義素子/基本素子へのポインタ
要素端子の個数 (n)
端子1へのポインタ
...
端子nへのポインタ

(c) : 端子

端子名
端子種別
配置座標
接続先端子へのポインタ
信号線へのポインタ
単位へのポインタ
データ型
L S B
O F F S E T

(d) 信号線

端子1へのポインタ
端子2へのポインタ
形状 (屈曲点の座標列)

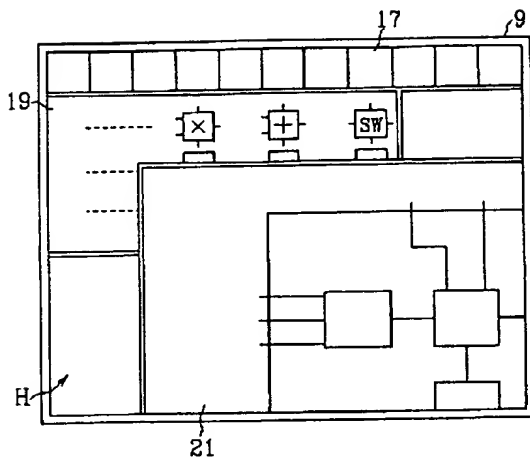
(e) : 単位

単位名
変換可能な単位の数 (n)
単位1の単位名
単位1への変換式
...
単位nの単位名
単位nへの変換式

(f) : 基本素子

基本素子名
端子の個数 (n)
端子1へのポインタ
...
端子nへのポインタ
シミュレーション動作
生成プログラム

【図 4】

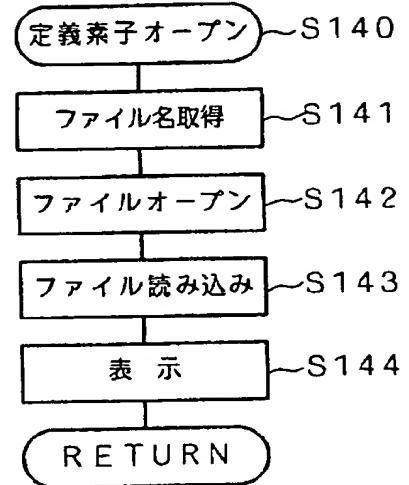


【図 30】

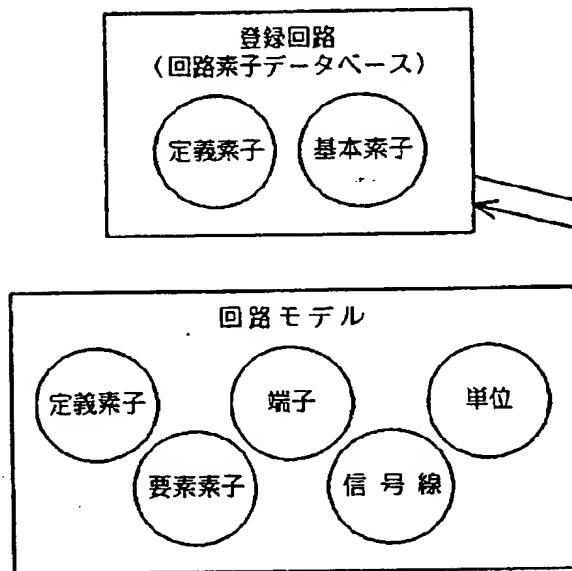
ヘッダファイル

```
#define T3() E5() -Hf1
#define A5() D3() -Hf2
#define T2() T1() -Hf3
#define A6() A4() -Hf4
#define D3() D4() -Hf5
#define A7() A1() -Hf6
#define A8() A2() -Hf7
#define A9() A3() -Hf8
#define A10() A7() -Hf9
#define A11() D5() -Hf10
#define A12() A8() -Hf11
#define A13() A9() -Hf12
```

【図 7】

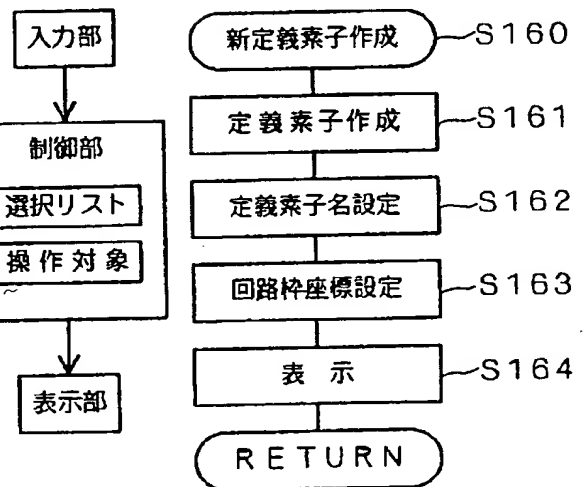


【図 5】

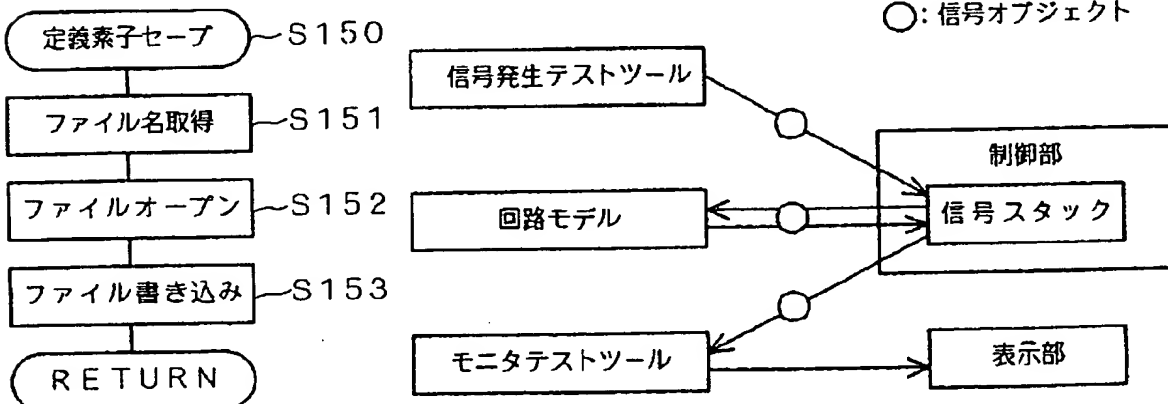


【図 8】

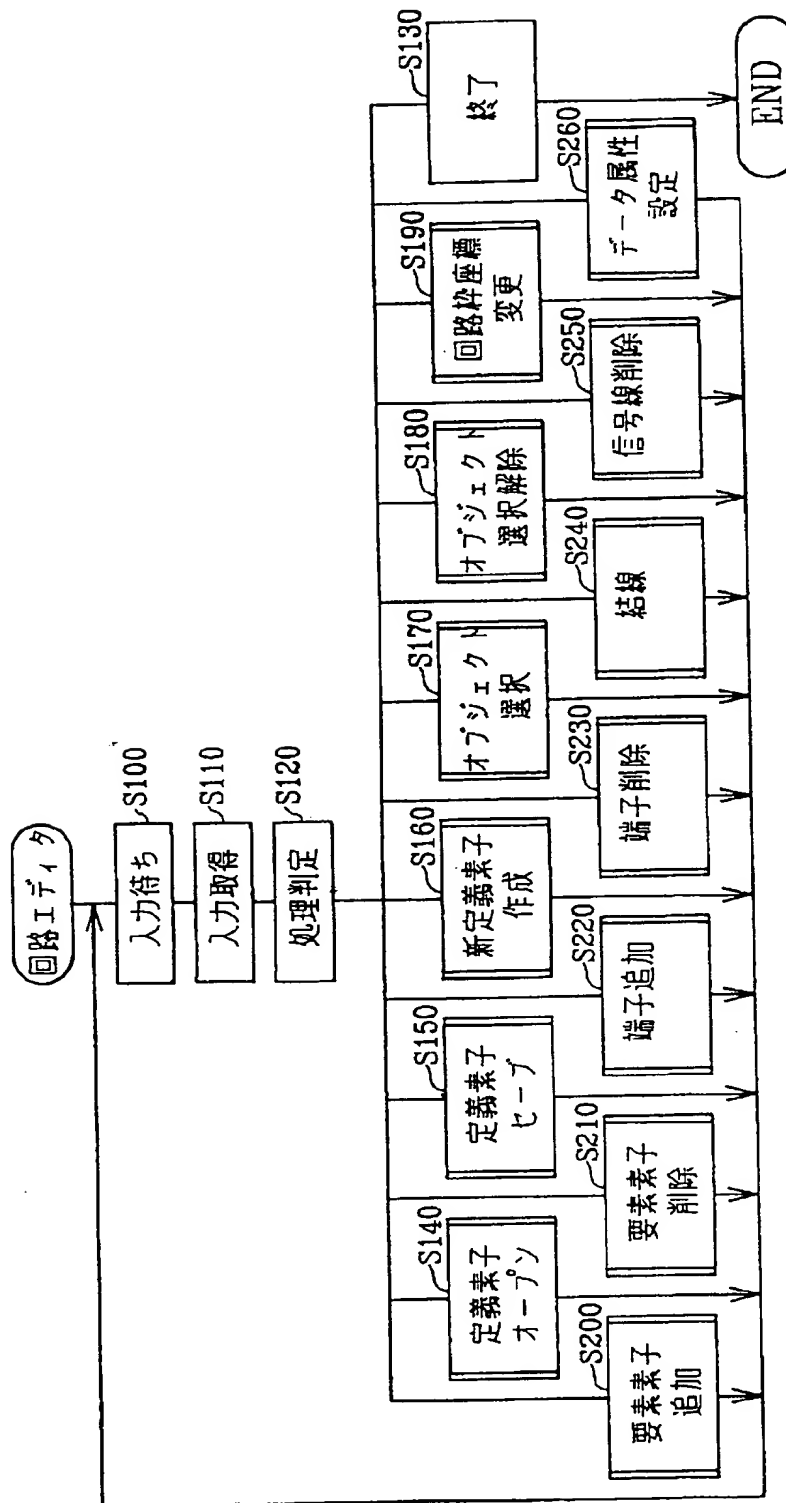
【図 9】



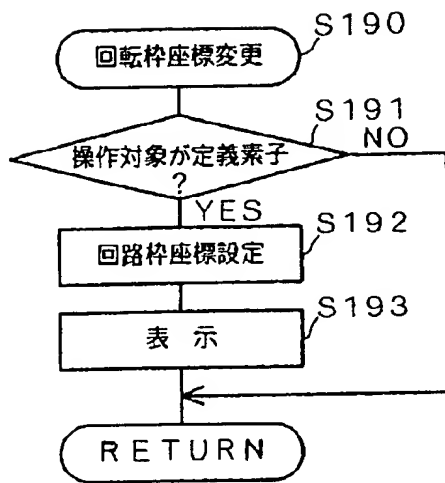
【図 22】



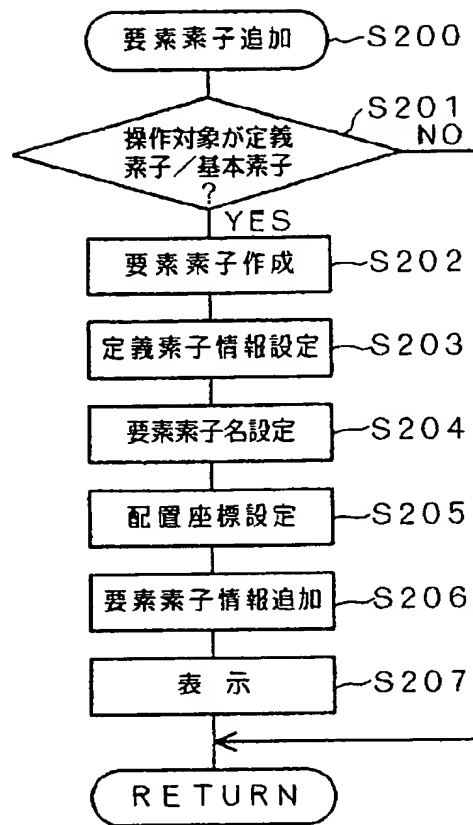
【図6】



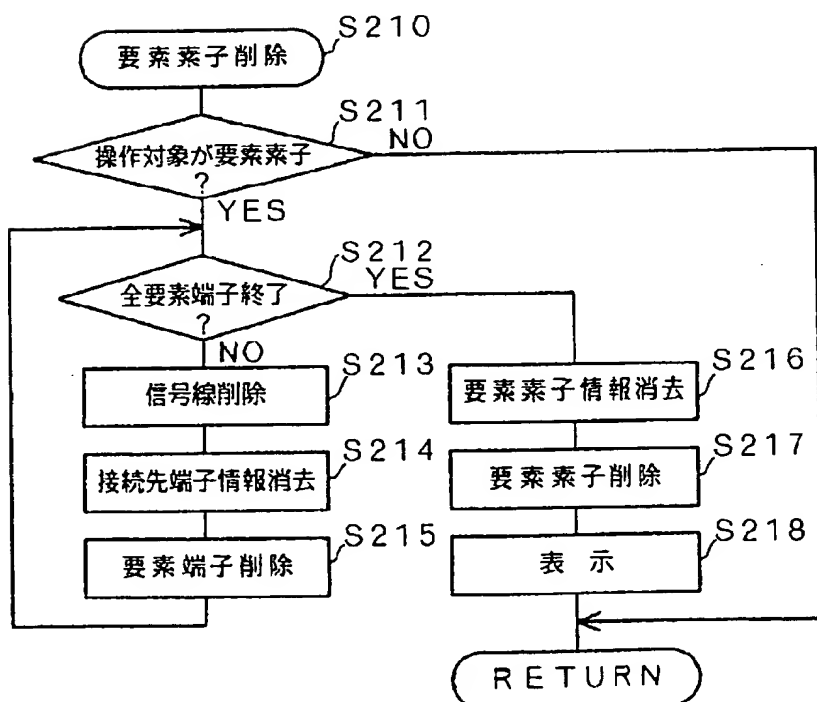
【図12】



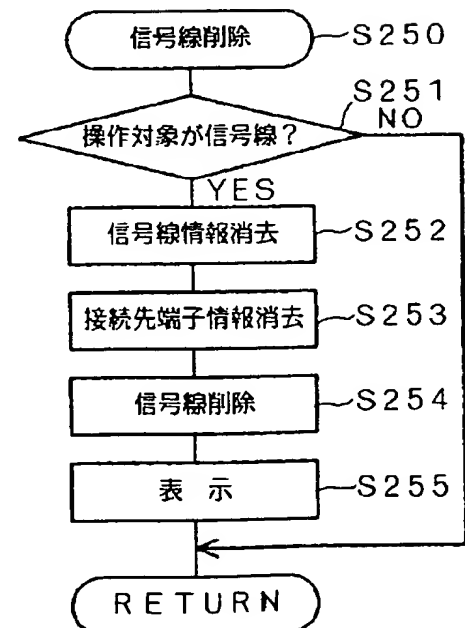
【図13】



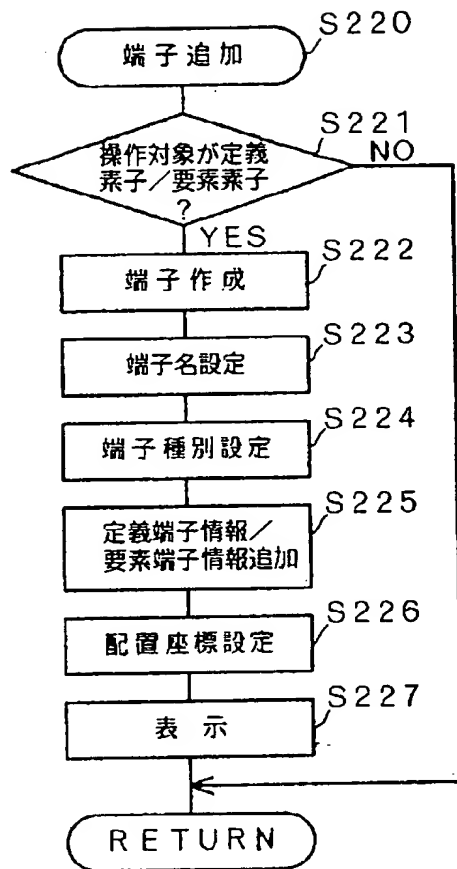
【図14】



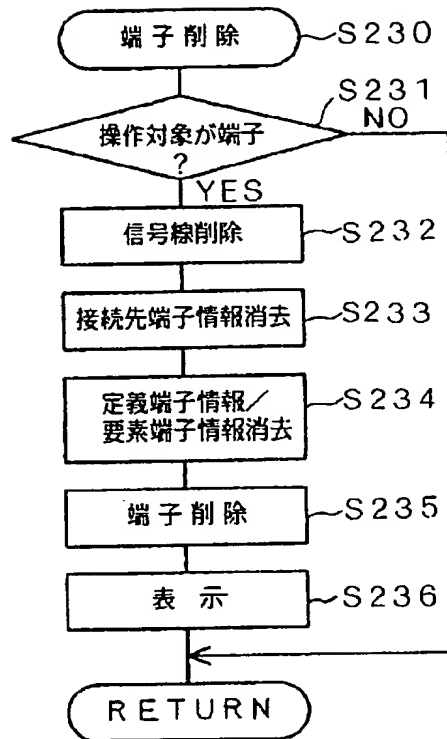
【図18】



【図15】



【図16】



【図20】

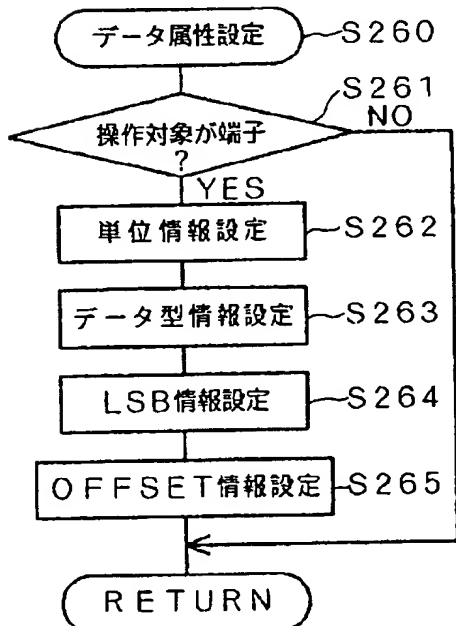
(a) : テストツール

テストツール名
端子の個数 (n)
端子1へのポインタ
...
端子nへのポインタ
シミュレーション動作

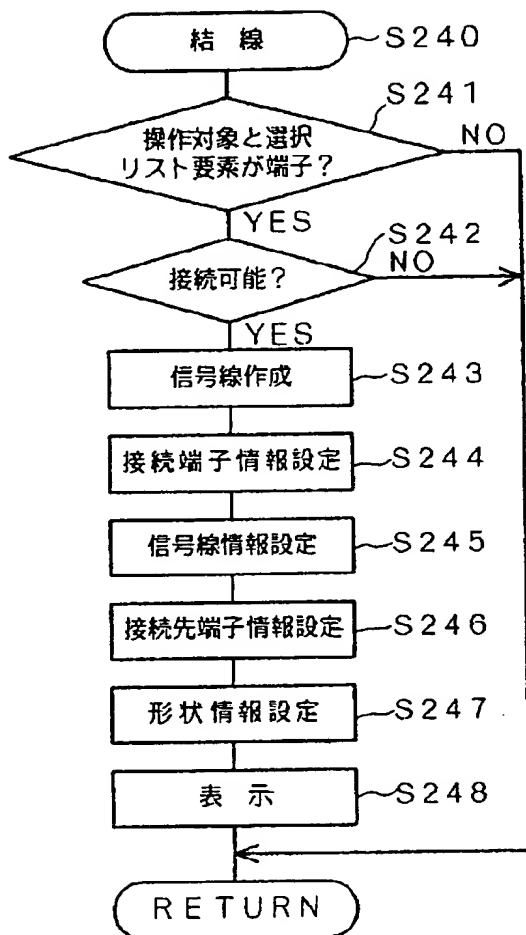
(b) : プローブ

端子1 (テストツール側端子) へのポインタ
端子2 (回路側端子) へのポインタ

【図19】



【図 17】

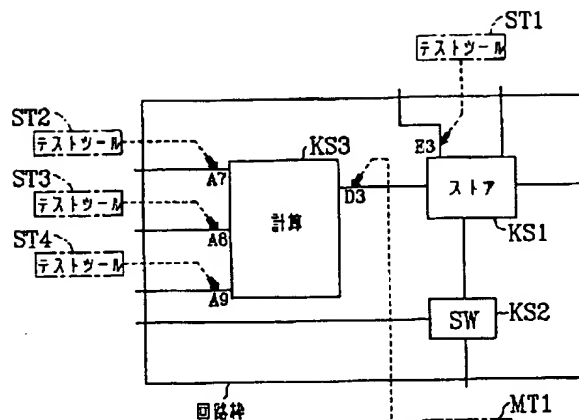


【図 23】

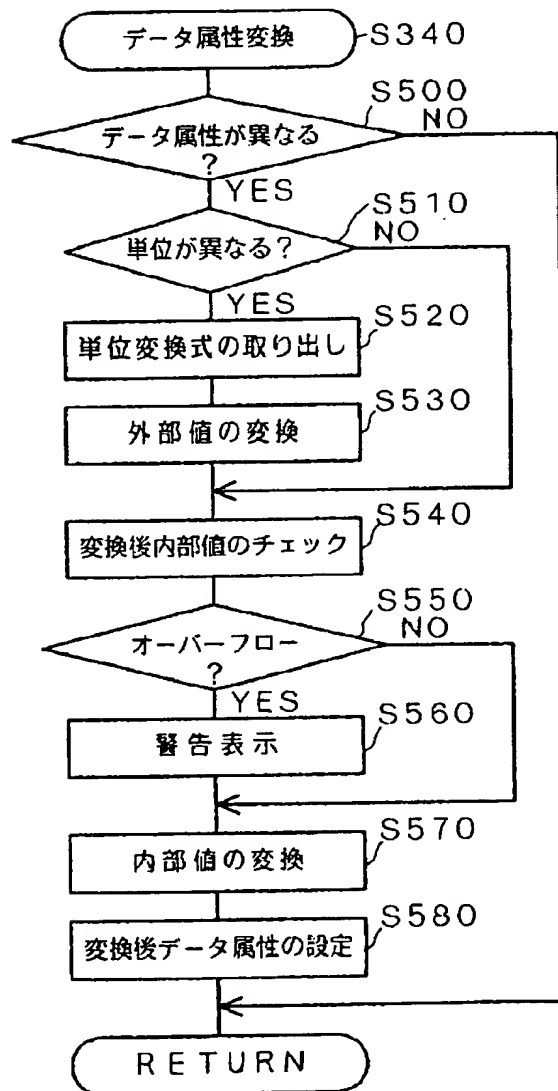
信号オブジェクト

信号種別
発信端子へのポインタ
受信端子へのポインタ
データの外部値 [データ信号のみ]
データの内部値 [データ信号のみ]
単位へのポインタ [データ信号のみ]
データ型 [データ信号のみ]
LSB [データ信号のみ]
OFFSET [データ信号のみ]

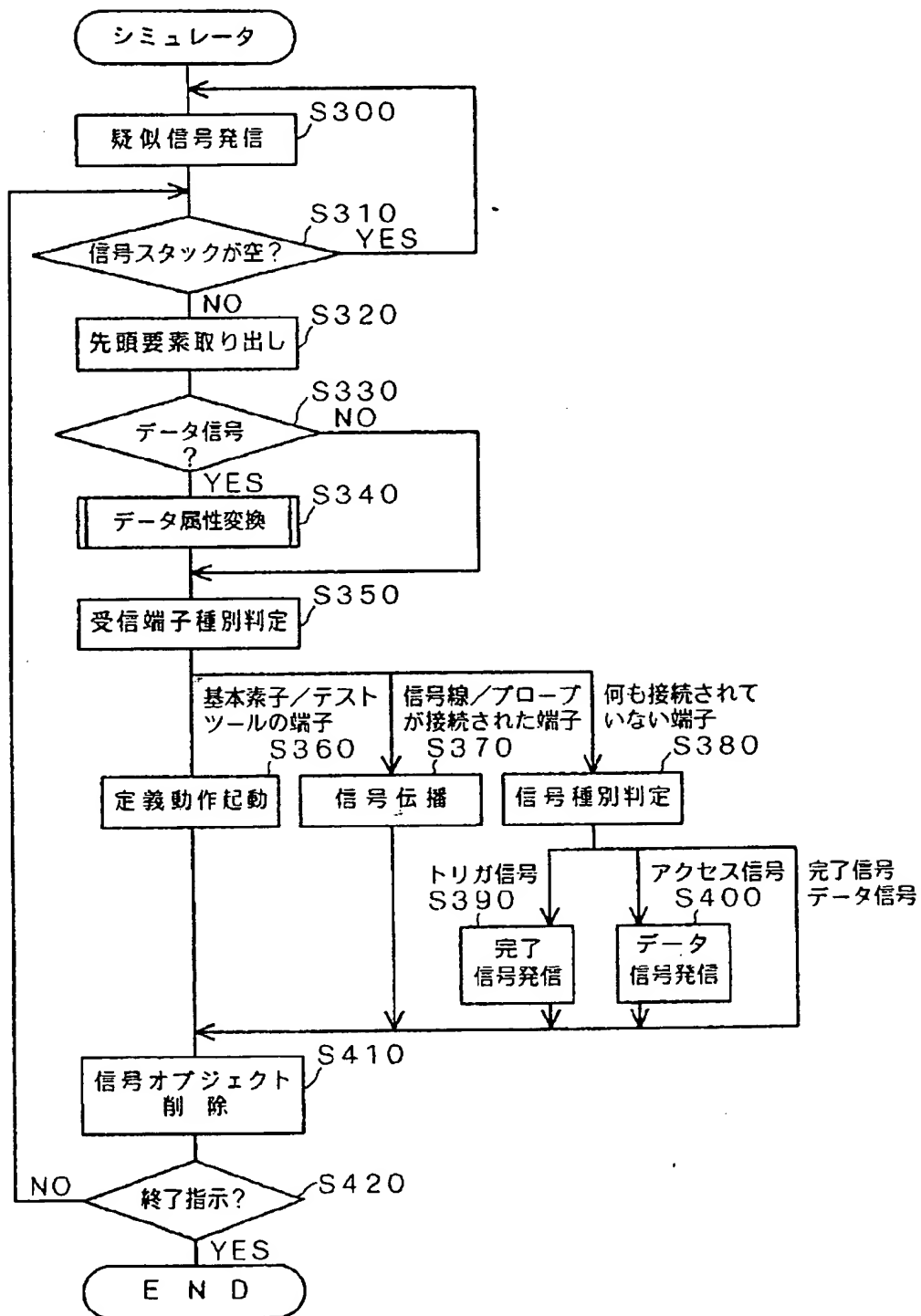
【図 21】



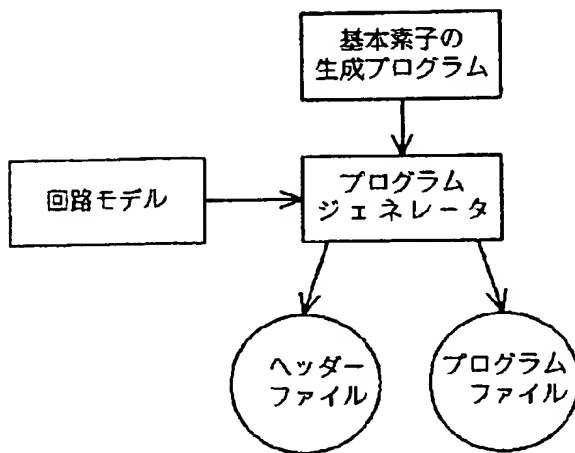
【図 25】



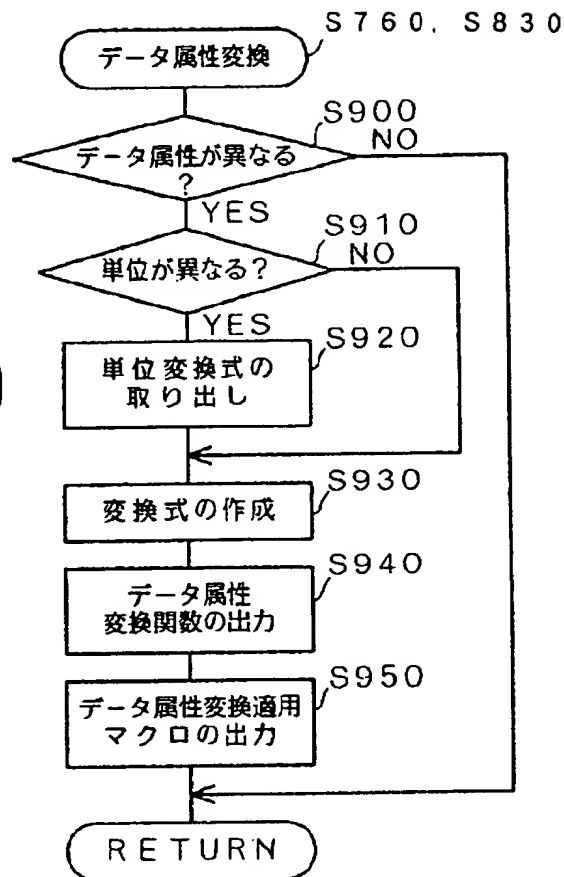
【図 24】



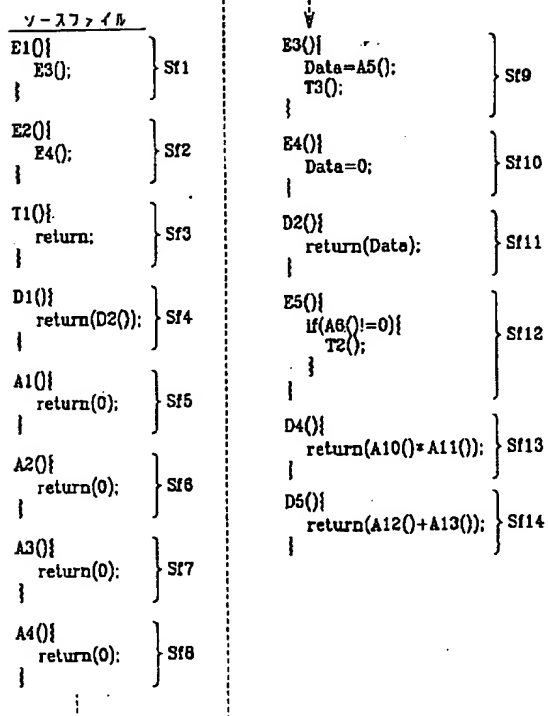
【図 26】



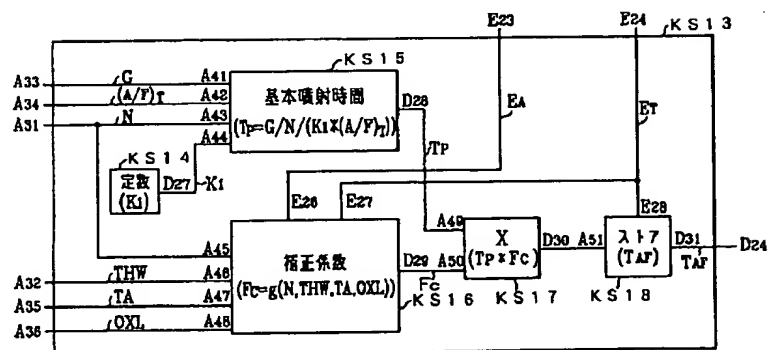
【図 29】



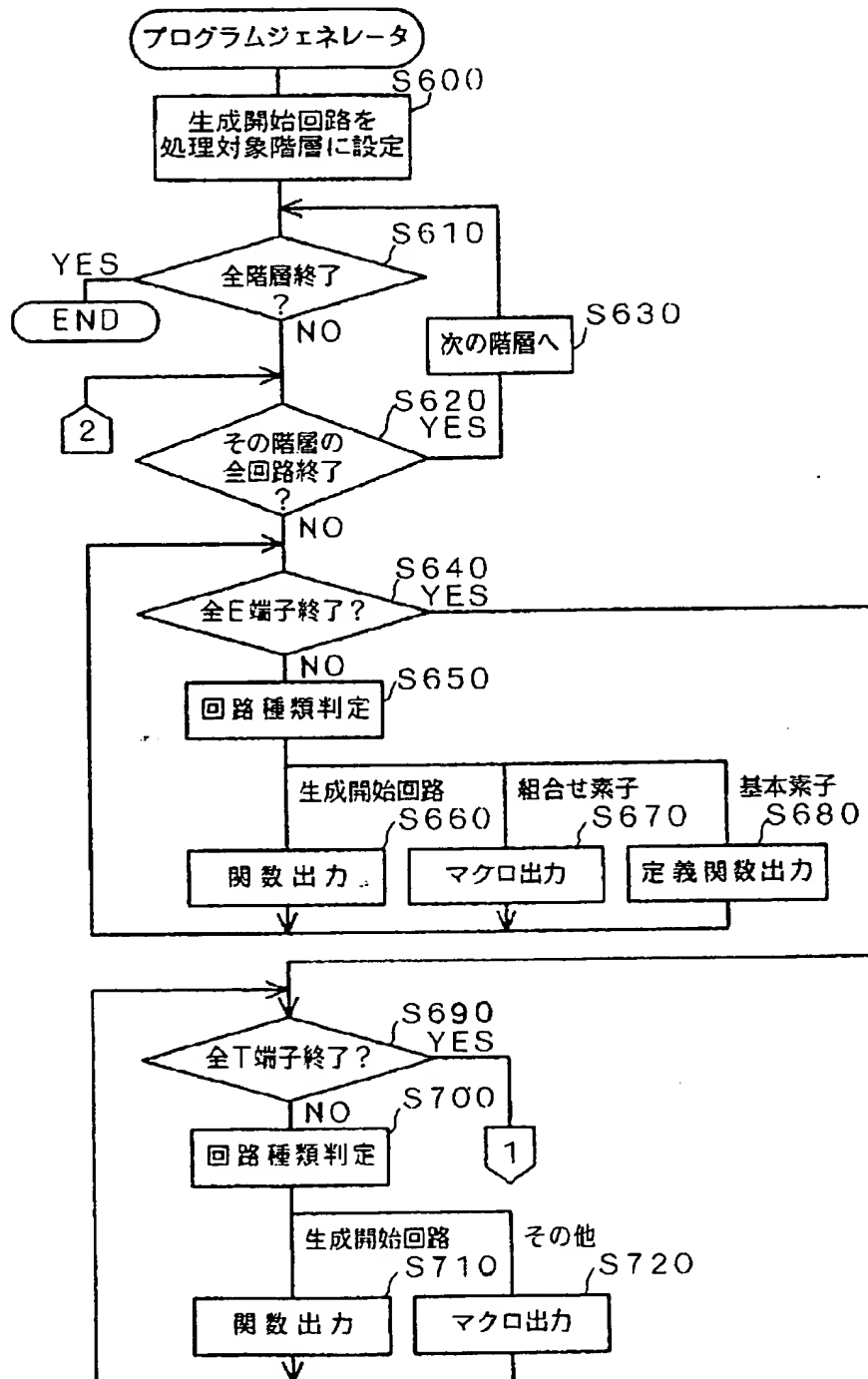
【図 31】



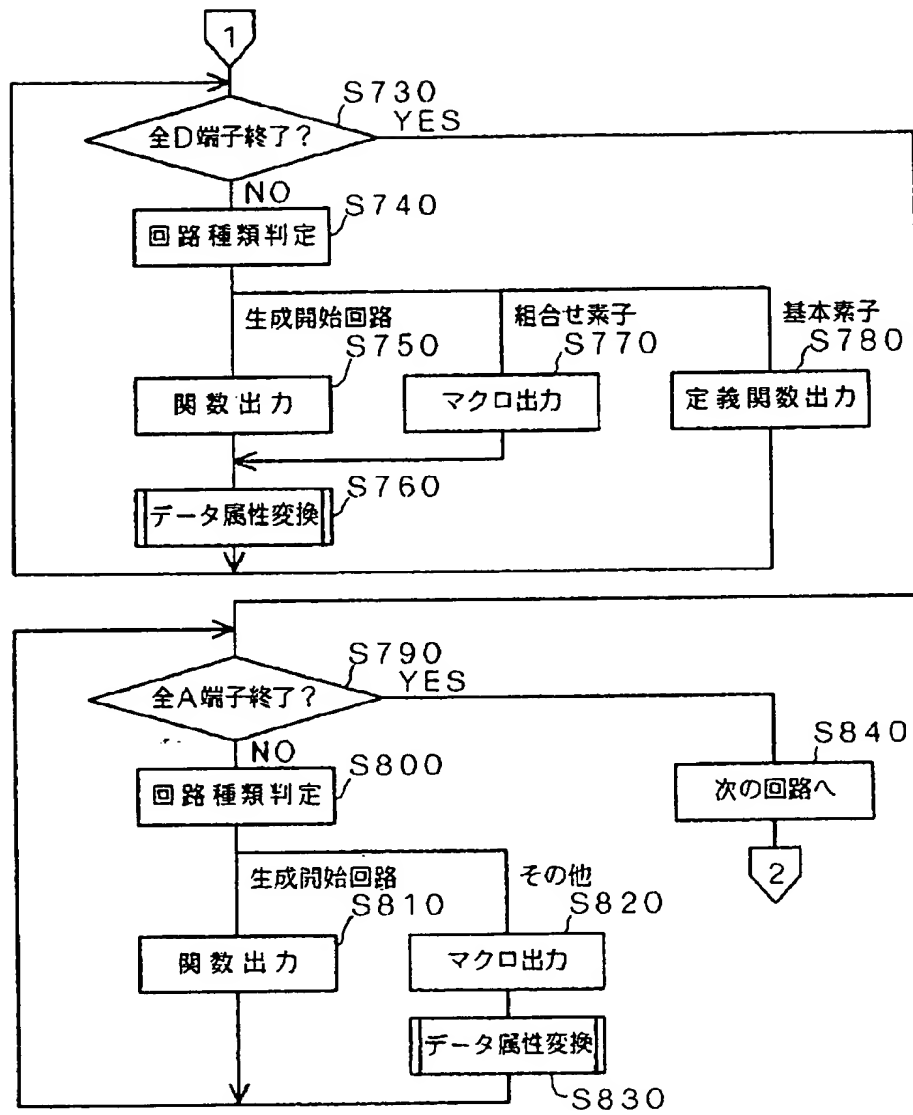
【図 35】



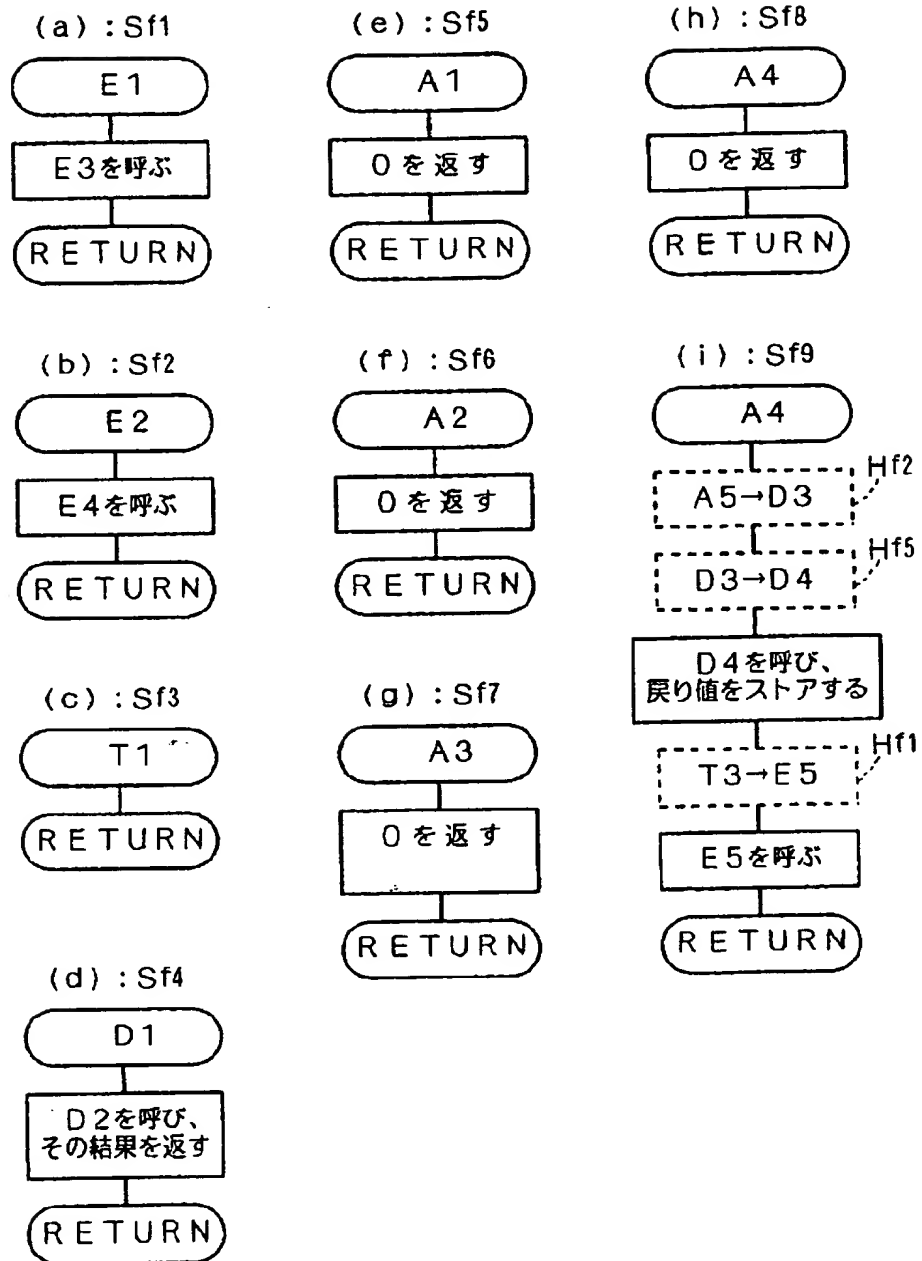
【図27】



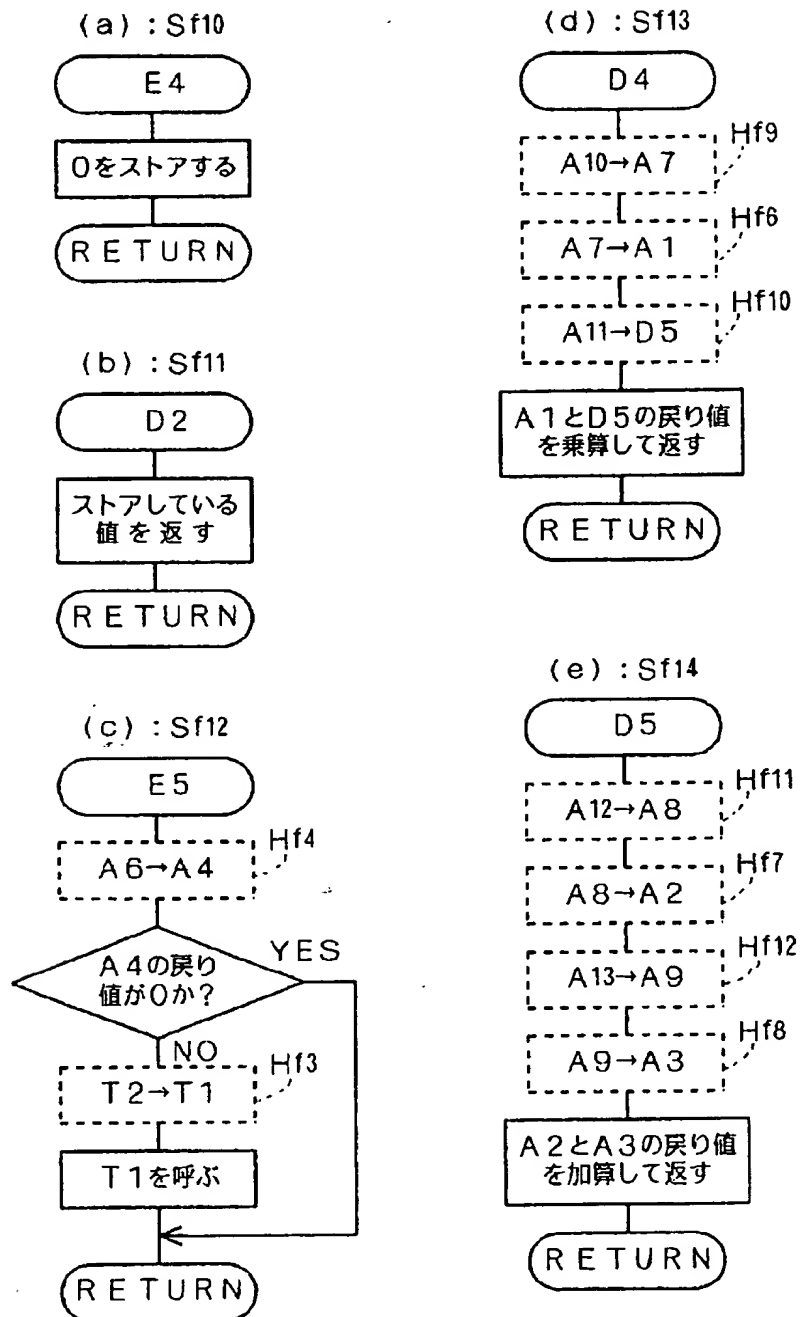
【図28】



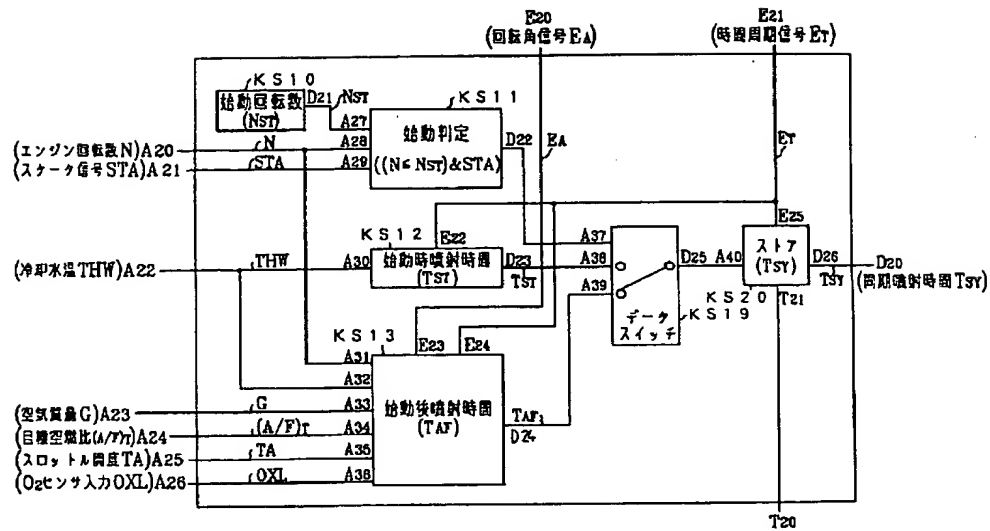
【図32】



【図33】



【図 34】



フロントページの続き

(72) 発明者 伊藤 健三
愛知県名古屋市中区錦二丁目15番20号 株
式会社デンソークリエイト内

(72) 発明者 藤井 友康
愛知県名古屋市中区錦二丁目15番20号 株
式会社デンソークリエイト内